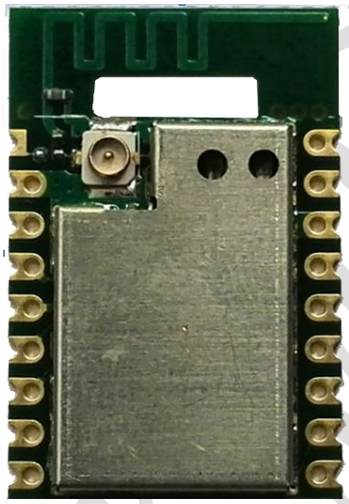


A6501

AirM2M_ESP8266_WiFi Module User Manual V2.9

Contents

- **Terms and acronyms**

Table 1 Terms and acronyms

Abbreviations	Descriptions
WiFi	Wireless Fidelity
AP	Access Point
UART	Universal Asynchronous Receiver & Transmitter
DTIM	Delivery Traffic Indication Message
SOC	System On a Chip
P2P	Point to Point
TCP	Transmission Control Protocol
IP	Internet Protocol
STBC	Space-Time Block Coding
MCU	Microprogrammed Control Unit
MIMO	Multiple Input Multiple Output
MPDU	MAC Protocol Data Unit
MSDU	MAC Server Data Unit
IOT	Internet Of Things
GPIO	General Purpose Input/Output
OTA	Over The Air
IEEE	Institute Of Electrical And Electronics Engineers
bps	Bits Per Second
CCK	Corporate Control Key
DQPSK	Differential Quadrature Phase Shift Keying
DBPSK	Differential Binary Phase Shift Keying
QAM	Quadrature Amplitude Modulation
OFDM	Orthogonal Frequency Division Multiplexing
WPA	Wi-Fi Protected Access
TKIP	Temporal Key Integrity Protocol
WAPI	Wlan Authentication And Privacy Infrastructure
WEP	Wired Equivalent Privacy
CRC	Cyclic Redundancy Check

1. Product introduction

1.1. Summary

Up to now Shanghai AirM2M Communication Technology Co., Ltd has released 3 kinds of ESP8266 WiFi modules, including A6501、A6501S and A6502. There will be more kinds coming to meet different PCB package requirements of enterprise users and individual developers.

A650X series modules own competitive PCB packaging sizes in the industry and ultralow power consumption technology. They are designed for mobile devices and IOT applications, which can connect users' physical devices to WiFi wireless network,so communications on INTERNET or LAN may happen.

There are different kinds of PCB packagings for A650X series modules. Some models of antennas are supported,including on-board PCB antenna, IPEX interface and stamp hole interface.

A650X series modules can be widely used on smart power grids, intelligent transportation, intelligent furniture, handheld devices, industrial control and so on.

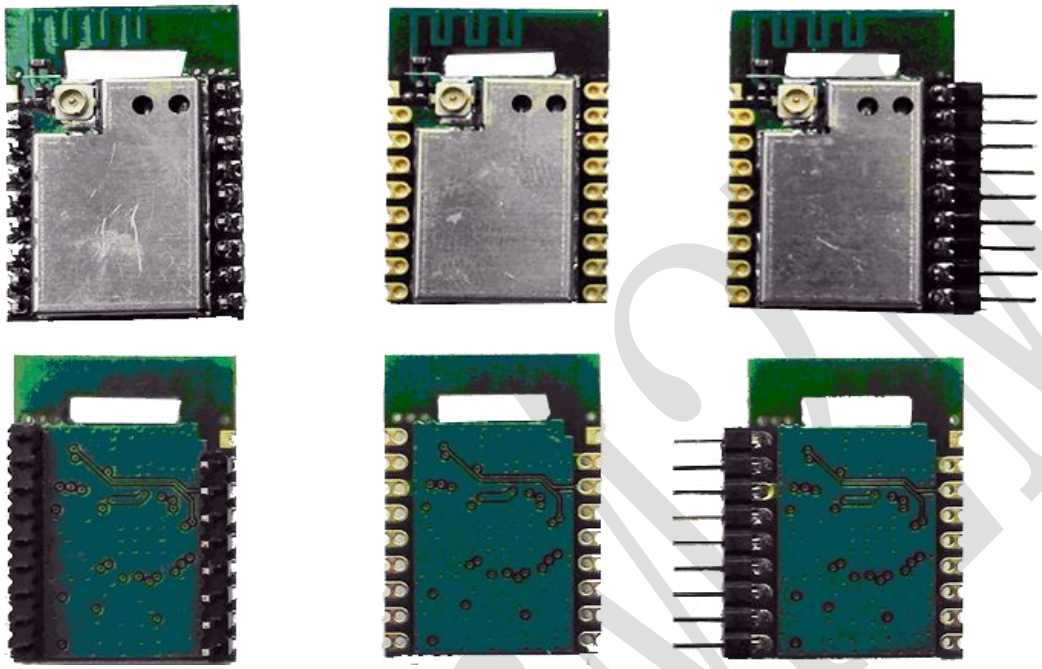
For more details please refer to http://www.luam2m.com/

1.1.1 Product features

- Support 802.11 b/g/n wireless standard
- Support STA/AP/STA+AP 3 working modes
- Built-in TCP/IP protocol stack, support multipath TCP Client connections
- Support abundant Socket AT commands.
- Built-in 32 bit MCU, can be used as the application processor
- Support UART/GPIO data communication interfaces
- Support PWM output
- Support I2C interfaces
- Support WiFi configuration parameters
- Support Smart Link intelligent networking function
- Support remote firmware upgrade（OTA）
- 3.3V single power supply
- Support multiple power consumption modes like Active、Modem sleep、Light sleep、Deep sleep
- Ultra low power consumption, suitable for battery-powered applications

1.1.2 Module Packages

AirM2M ESP8266 A650X series modules support multiple PCB packages, which greatly enriches users' choices. They can be easily applied to a variety of networking applications and solutions.



Picture 1 front & bottom view of A6501 module

1.1.3 Specifications

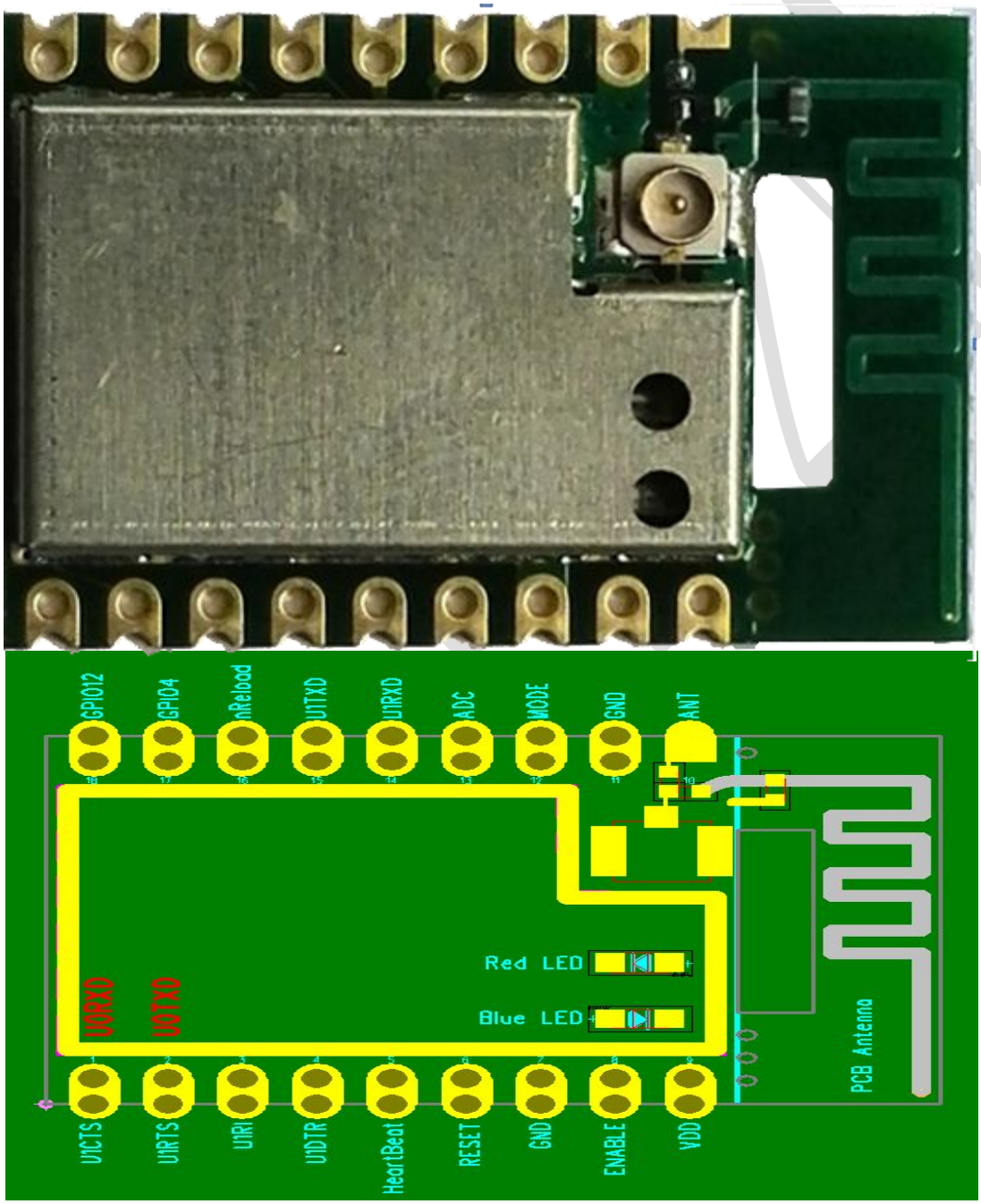
Table 2 A6501 Module technical specifications

A650X	Mater chip	ESP8266	
Wireless parameters	wireless standard	IEEE 802.11b/g/n	
	frequency range	2.412GHz-2.484GHz	
	transmitting power	802.11b: +18.5 +/-1dBm (@11Mbps)	
		802.11g: +16 +/-1dBm (@54Mbps)	
		802.11n: +15 +/-1dBm (@HT20, MCS7)	
	receiving sensitivity	802.11b: -91 dBm (@11Mbps ,CCK)	
		802.11g: -75dBm (@54Mbps, OFDM)	
		802.11n: -71dBm (@HT20, MCS7)	
	antenna mode	built-out: stamp hole interface or I-PEX connector	
		built-in: onboard PCB antenna	
Hardware parameters	hardware interface	UART，IIC，PWM，GPIO，ADC	
	working voltage	3.0V--3.6V(VDD voltage)	
	GPIO drive capability	Max：15ma	
	working current	normal mode	average value：~70mA, peak value: 215mA
		Deep Sleep	<15uA
		Modem Sleeps	average value： ~15mA
		Light Sleep	0.9mA
	operating temperature	-40℃ ~ 125℃	
	storage environment	temperature：<40℃，relative humidity：<90%R.H.	
	dimension	onboard PCB antenna：16*24*3.3mm； built-out antenna：16*18.6*3.3mm	
Serial transmission	transmission rate	9600 ~ 921600 bps	
	TCP Client	4	
Software parameters	wireless network types	STA/AP/STA+AP	
	security mechanism	WEP/WPA-PSK/WPA2-PSK	
	Encryption type	WEP64/WEP128/TKIP/AES	
	firmware upgrade	local serial, OTA remote upgrade	
	networking protocol	IPv4, TCP/UDP/HTTP	
	user configuration	AT+ instruction set Web page + PC configuration tool Android/iOS terminal Smart Link intelligent configuration APP	

1.2. Hardware introduction

A650X series modules powered by ESP8266 SOC processor with abundant hardware interfaces support UART，IIC，PWM，GPIO，ADC, which can be widely used on different internet applications.

1.2.1 A6501 PIN Definition



Picture 2 A6501 pin configuration

Table 3 A6501 detailed Pin descriptions

Pin	Function	Description
1	U1CTS	1）UART_CTS (in normal mode, i.e. MODE pin float when booting up); 2）GPIO3(can be redefined in SDK firmware version); 3）U0RXD for ESP8266 firmware dowloading (MODE pin pulled down when booting up).
2	U1RTS	1）UART_RTS (in normal mode, i.e. MODE pin float when booting up); 2）GPIO1(can be redefined in SDK firmware version); 3）U0TXD for ESP8266 firmware dowloading (MODE pin pulled down when booting up); Note:Internally pulled up,please remain float when booting up, no pulling down!
3	U1RI	1）UART_RI, to wake up MCU, active low; 2）GPIO5(can be redefined in SDK firmware version);
4	U1DTR	1）UART_DTR,to wake up A6501, active low; 2）GPIO14(can be redefined in SDK firmware version);
5	HeartBeat	1）Wake up A6501(with AT+AMSLEEP) from deep sleep mode when shorted with RESET; 2）GPIO16(can be redefined in SDK firmware version); 3）High level output is defaulted after boot- up.
6	RESET	1）Non-deep-sleep mode: external reset signal, active low; 2）Deep sleep mode: to wake up A6501 when shorted with Heartbeat
7	GND	GND
8	ENABLE	1)Module enable Pin, internal 10K ohm pull up, powered on by VDD by default; 2)Shutdown on low level, power consumption current is less than 5uA
9	VDD	3.3V,module power supply
10	ANT	WiFi Antenna
11	GND	GND
12	MODE	1) Red LED:TCP/UDP connection state indicator. 2) GPIO0(can be redefined in SDK firmware version); 3) mode selection: Float when booting up：normal operating mode Pulled down when bootingup：firmware download mode
13	ADC	ADC, input range：0V-1V；
14	U1RXD	1）UART_RXD, receive data 2）GPIO13(can be redefined in SDK firmware version)
15	U1TXD	1) UART_TXD, send data 2) GPIO15(can be redefined in SDK firmware version); Note: internally pulled down at 10K ohm, no hardware pull up outside when booting up!
16	nReload	1) Press-key functions(“press” means pulling down the pin): press once:start Smartlink process; press twice:start OTA firmware download process; long press:reset to factory configuration. 2) GPIO2(can be redefined in SDK firmware version); 3) Debug_TXD, used for output of debugging log; AT+OPENLOG can open the log output function for this pin,which means the press-key funtions will cease to be effective. Note: Must kept high level when booting up, no hardware pull down please!

17	GPIO4	GPIO4(can be redefined in SDK firmware version);
18	GPIO12	1) Blue LED： TCP/UDP indication light control signal is defaulted. 2) GPIO12(can be redefined in SDK firmware version)

Notes:

1)There are 3 types serial ports(uarts),among which:

- UART0(U0RXD、U0TXD)is an uart for firmware download. Marked in red in table3.
- UART1(U1RXD、U1TXD、U1CTS、U1RTS、U1DTR、U1RI) is an uart for AT commands /data.
- Debug_UART(Debug_TXD)is a debug uart for log printing. Marked in dark yellow in table 3.

2) Please pay close attention to the levels of certain pins when the module is being powered on to avoid the malfuncition happening.

Pins that request initialization level when booting up are already marked by blue color.

Take a normal scenario for example(AT command input/output after power-on):

- step1:

The module is in power-off state;

- step2:

Keep MODE pin float, connect AT uart(i.e. data uart ,PIN14 and PIN15) with MCU;

- step3:

The module's flow control mode is OFF by default, this mode can be set to HARDWARE by AT+IFC,which means U1CTS(pin1) and U1RTS(pin2) need to be connected with MCU(if flow control is needed) or be shorted with each other(if flow control is not needed).

- step4:

Power on the module,pay close attention that pin2 and pin16 are forbidden to be pulled down and pin15 is forbidden to be pulled up at the moment of power-on.

- step5:

Input AT commands via AT uart. The baud-rate is 921600 by default , and baud-rate lower than 921600 can be trained by inputting several AT commands(please refer to 5.1.2 Set Uart baud rate：AT+IPR for more details)

3) PIN "MODE" is used for module mode selection. Connect "MODE"to"GND" and then power on the module,the module will go into firmware upgrading mode.

4) Statements for"Red LED" and "Blue LED"：

Red LED：

As the TCP/UDP connection state indicator, blinks when connected, extinguishes when disconnected.

Blue LED：

a) Lit for 2S,and then turned off after boot-up；

b) As the TCP/UDP data transmission state indicator ,blinks when transmitting, extinguishes when transmission finished.

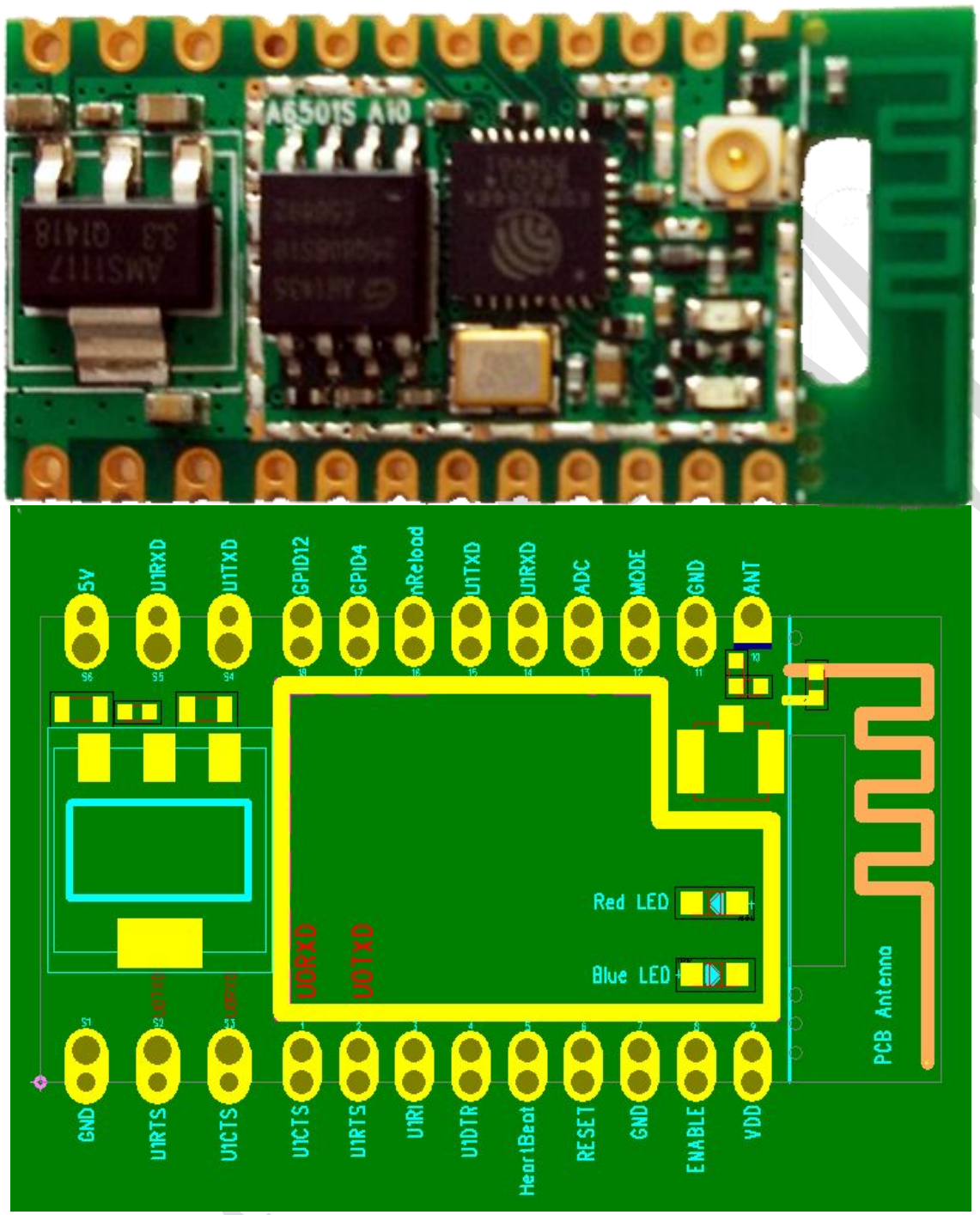
5) There are 3 GPIO pins of A6501 which are completely unrestricted (no request on initialization

level, no default function or default function can be modified). They are marked by green. First using these pins is recommended.

6) Special reminding：

Please remember that in firmware versions later than **V0003**, firmware upgrade serial and data transmission serial do not use the same pins any more.

1.2.2 A6501S



Picture 3 A6501S pin configuration

For the 18 fixed pins, A6501S and A6501 are totally identically defined. The newly added pins S1---S6 ,

except “5V”, are defined in the same way as 5 of the fixed pins but the space between the added pins is 2.54mm. The purpose of leading out the 6 pins that are often used is to make it easy for debugging.

Details are as following:

PIN	Function	Description
S1	GND	GND
S2	U1RTS	identical to PIN2
S3	U1CTS	identical to PIN1
S4	U1TXD	identical to PIN15
S5	U1RXD	identical to PIN14
S6	5V	module power supply

Table 4 A6501S S1---S6 Pin definition

Main distinctions between A6501S and A6501 are：

(1) One LDO is added： AMS1117, developer can use the 5V power of serial port line USB2TTL directly for power supply ;

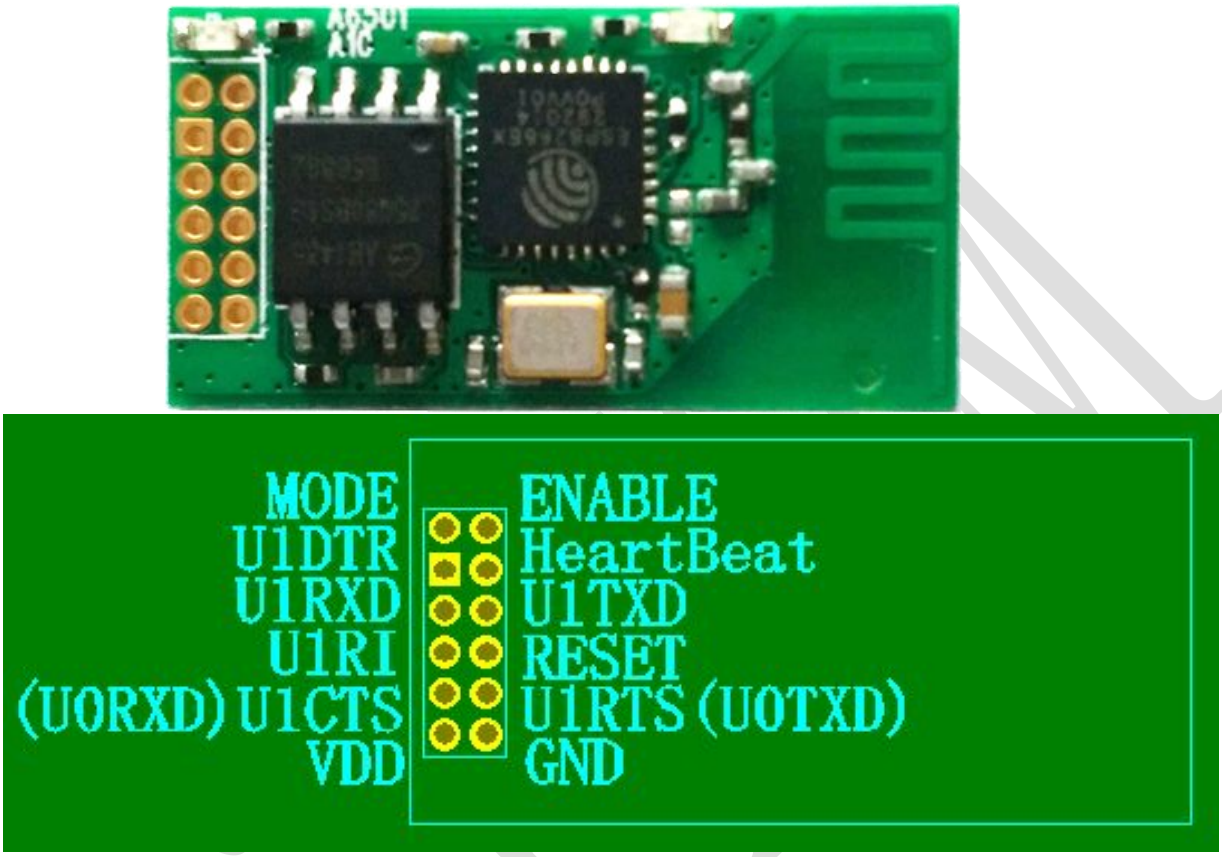
(2) 6 pins are added whose space between each other is 2.54mm：5V、GND、U1RXD、U1TXD、U1RTS、U1CTS;

(3)Shielding case is not welded in the developer’s versions of modules for easy developments.

(4)The dimension is added from 16mm*24mm*3.3mm（A6501）to 16mm*32mm*3.3mm（A6501S）;

(5)A6501S+USB2TTL Serial port line will be enough for debugging directly, no need for development board.

1.2.3 A6502



Picture 4 A6502 pin configuration

A6502 uses 12 pins whose spacing is 1.27mm leading out all the signals that are needed for serial port data transmission application. Compared to A6501 和 A6501S, the main characteristics of A6502 are as follow:

(1) smaller dimension：11.8mm*24mm*3.0mm；

(2) Pin spacing has been changed to 1.27mm；

(3) Hardware all flow control interface can be applied to various data transmission applications.

1.3. Power consumption

All the following power consumption data come from tests basing on 3.3V power supply and 25° C environment temperature.

[1] All the tests are completed at the antenna interface.

[2] All emitting data are acquired in continuous emission mode, basing on a duty ratio of 90%.

Mode	Min	Normal	Max	unit
transmit 802.11b，CCK 1Mbps，Pout=+19.5dBm		215		mA
transmit 802.11b，CCK 11Mbps，Pout=+18.5dBm		197		mA
transmit 802.11g，OFDM54 Mbps，Pout=+16dBm		145		mA
transmit 802.11n，MCS7，Pout=+14dBm		135		mA
receive 802.11b，package length is 1024 bytes，-80dBm		60		mA
receive 802.11g，package length is 1024 bytes，-70dBm		60		mA
receive 802.11n，package length is 1024 bytes，-65dBm		62		mA
System standby mode		0.9		mA
Deep sleep		10		μA
Energy-saving mode DTIM1		1.2		mA
Energy-saving mode DTIM3		0.86		mA
Shutdown		0.5		μA

Table 5 Power consumption data

1.4. Radio frequency index

Following data gained at indoor temperature, voltage 3.3V.

Description	Min	Normal	Max	Unit
Input frequency	2412		2484	MHz
Input resistance		50		Ω
Input reflection			-10	dB
Under 72.2Mbps， PA ’s output power	14	15	16	dBm
In 802.11b mode， PA’s output power	17.5	18.5	19.5	dBm
Sensitivity				
CCK 1Mbps		-98		dBm
CCK 11Mbps		-91		dBm
6Mbps(1/2BPSK)		-93		dBm
54Mbps(3/4 64-QAM)		-75		dBm
HT20，MCS7（65Mbps，72.2Mbps）		-71		dBm
Adjacent channel suppression				
OFDM，6Mbps		37		dB
OFDM，54Mbps		21		dB
HT20，MCS0		37		dB
HT20，MCS7		20		dB

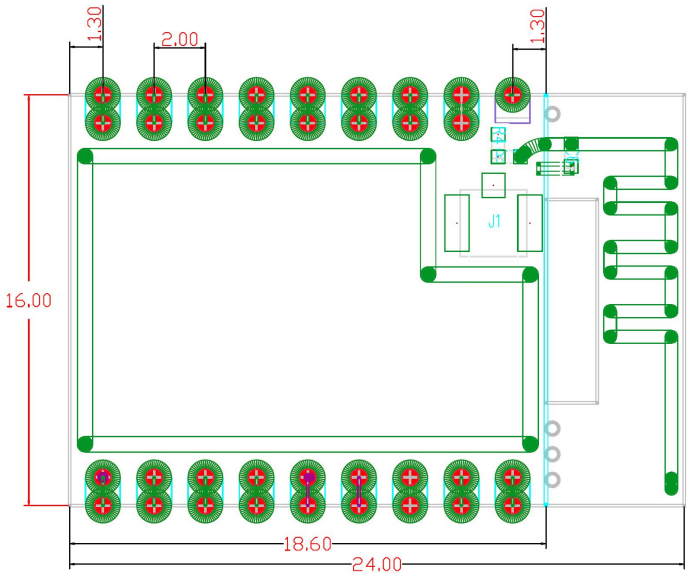
Table 6 Radio frequency index

Note：

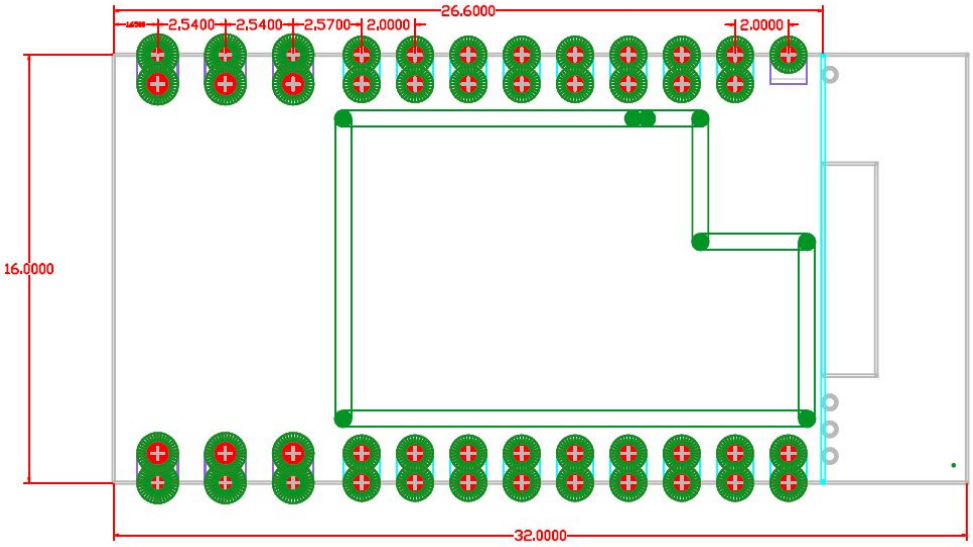
1）72.2Mbps gained in 802.11n mode, when MCS=7， GI=200uS.

2）Output power can reach to +19.5dBm in 802.11b mode.

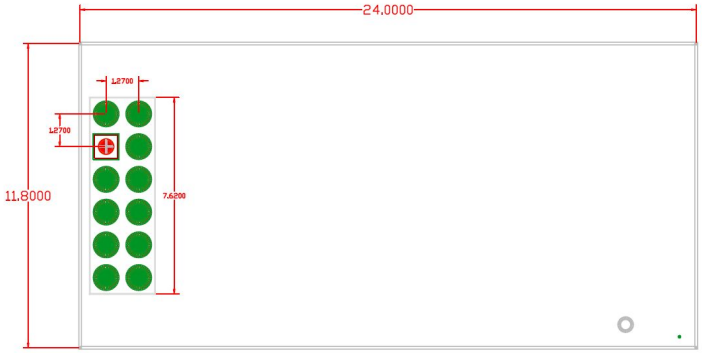
1.5. Dimension



A6501



A6501S

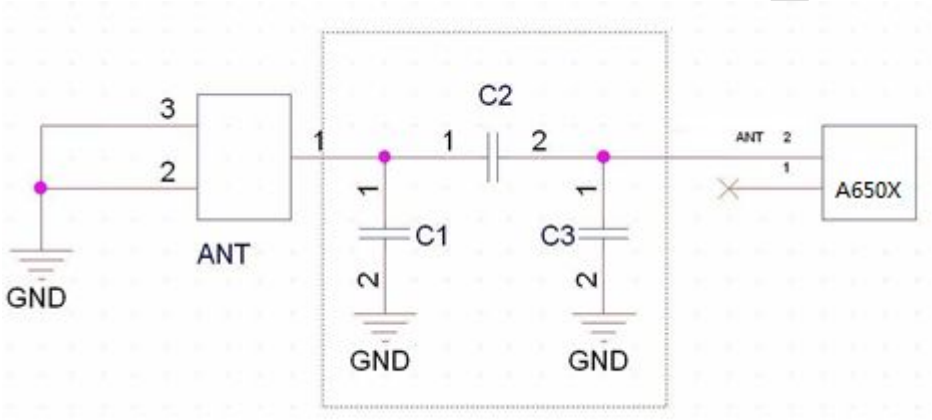


A6502

Picture 5 A650X module Pin Design

1.6. WiFi antenna

A650X series modules support 3 kinds of antenna interfaces. They are onboard PCB antenna, IPEX interface and stamp hole interface. Customers can use onboard PCB antenna and IPEX interface directly without adding any matching circuit. Customers can use A650X stamp hole antenna interface if they want to design antenna part on big board. In this case, matching circuit should be reserved on big board for this design. For example:



Picture 6 WiFi radio frequency reference circuit

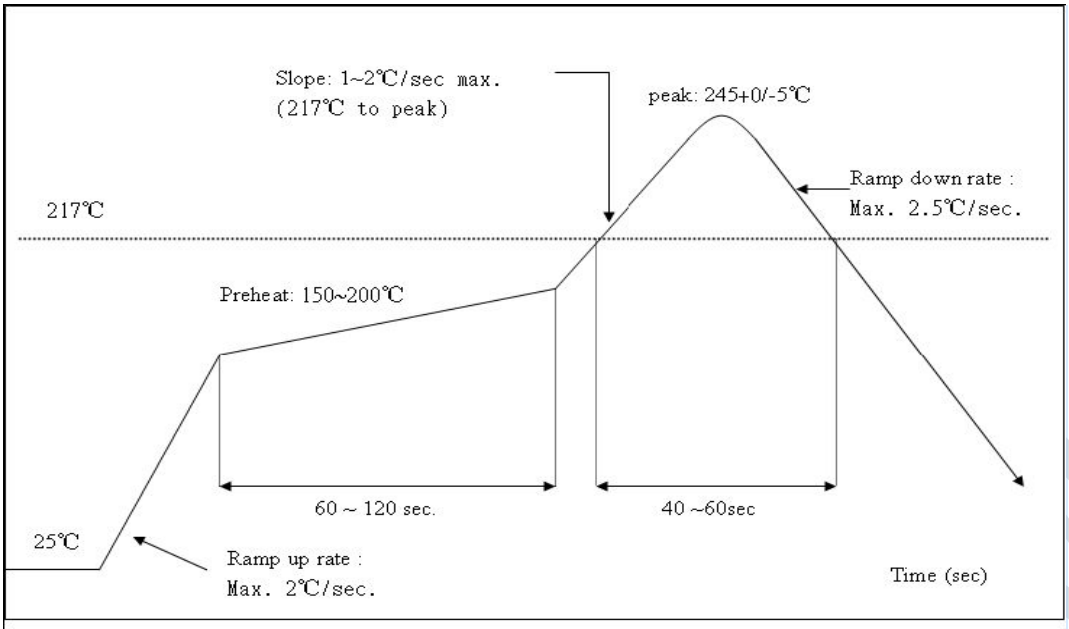
Note:

1、 The parts inside the above dashed box need antenna matching. Should take actual electronic parameters of the matching components of the antenna as standard.

2、 Need 50 ohm resistance for RF wiring. 90 degree angle wiring is forbidden. No longer than 15mm.

1.7. Recommended furnace temperature curve

Refer to IPC/JEDEC standard ； Peak Temperature : <250°C ； Number of Times: ≤2 times ；



Picture 7 Recommended back flow curve

2. Functional description

2.1. Main function

Main functions that A650X series modules can achieve include serial port transmission, PWM control and GPIO control.
Serial port data transmission: good reliability, stability, can reach 11Kbyte at serial rate 115200kbps
PWM control: light-control, three-colour LED control, motor speed control
GPIO control: control switch, relay and so on.

2.2. Operating mode

A650X modules support STA/AP/STA+AP 3 operating modes.

- STA mode: A6501 modules can connect to the internet by router. So cell phone or computer can remote control device by internet.
- AP mode: As hotspot, A6501 modules achieve that cell phone, computer can communicate with modules directly, which makes local area network wireless connections come true.
- STA+AP mode: Coexistence of these two modes.

2.3. Application fields

- RS232 serial data to Wi-Fi interface
- industrial data transparent transmission
- Wi-Fi remote control
- toy field
- color LED control
- fire protection, security and intelligent integrated management
- Intelligent card terminal, wireless POS machines, Wi-Fi cameras, handheld devices.

2.4. AirM2M Cloud

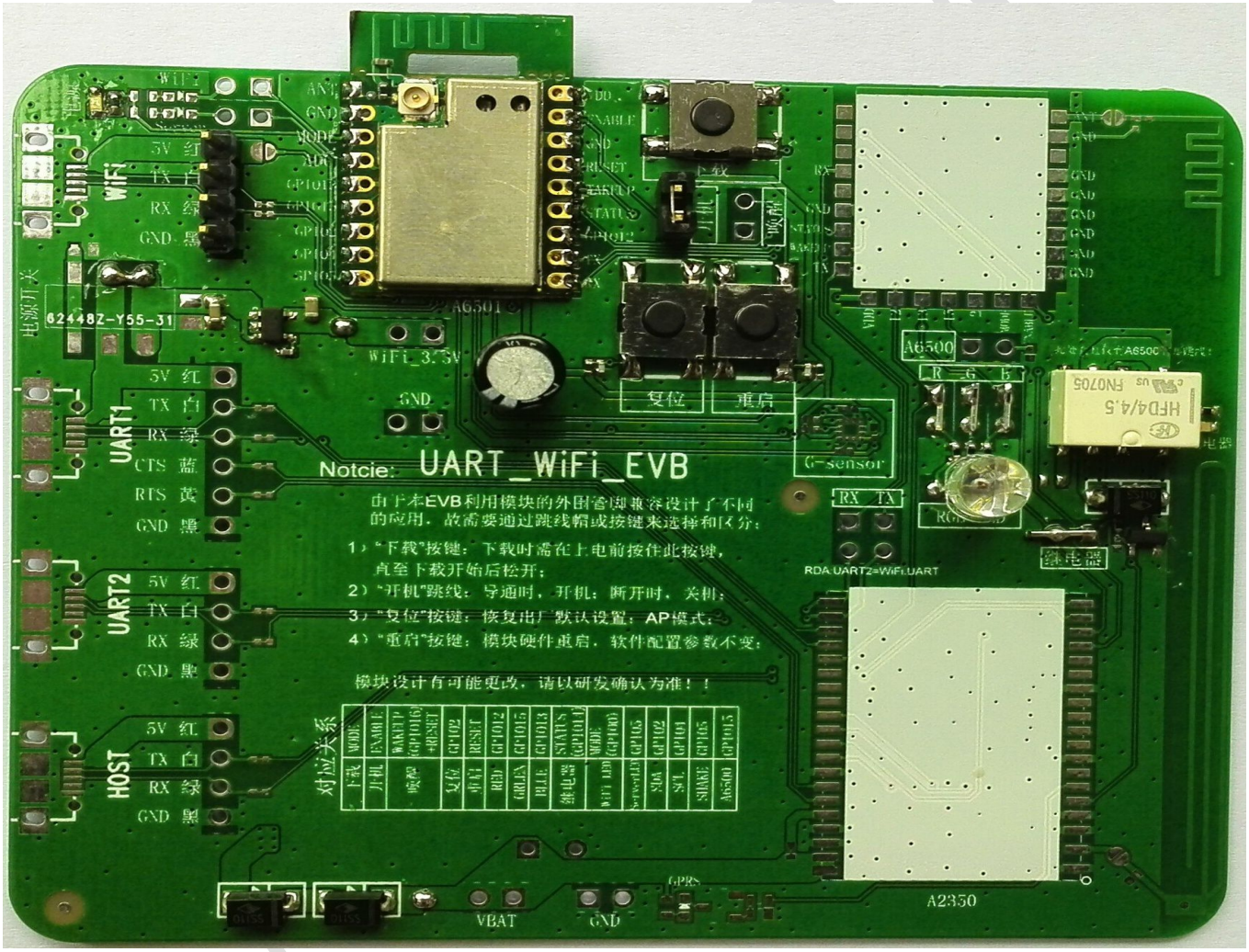
AirM2M Cloud is internet cloud platform service provided by Shanghai AirM2M（AirM2M）. Users can monitor and manage device on the platform to realize huge data management and analysis, which make device intelligence really come true.

AirM2M Cloud can accept customers' customized requests, including Web Page Configuration, Android/iOS platform and App development.

3. EVB Introduction

AirM2M can offer special UART_WiFi_EVB development board for customers' to debug A650X. By this development board, traditional serial device or MCU device can easily connect to WiFi network to realize managing and controlling of the devices by network.

This development board can offer hardware demonstration programs like UART serial port data transmission solution, RGB light-control and intelligent socket. At the same time, development board has reserved hardware circuit on which our GSM/GPRS standard module A2350 works as main controlling unit(MCU) to communicate with A6501 via serial port. So it can support GSM/GPRS long-distance transmission, and support LAN range WiFi communication.



Picture 8 EVB front view

Notice：

Because this EVB is compatible to different application by using module's periphery pins. So it need jumper cap or buttons for distinguishing.

1) Button "download": Press this button before power up for download until downloading begins.

2) Jumper wire "boot": when break over, turn on; when break, shutdown.

3) Button "reset": Restore factory default settings: AP mode.

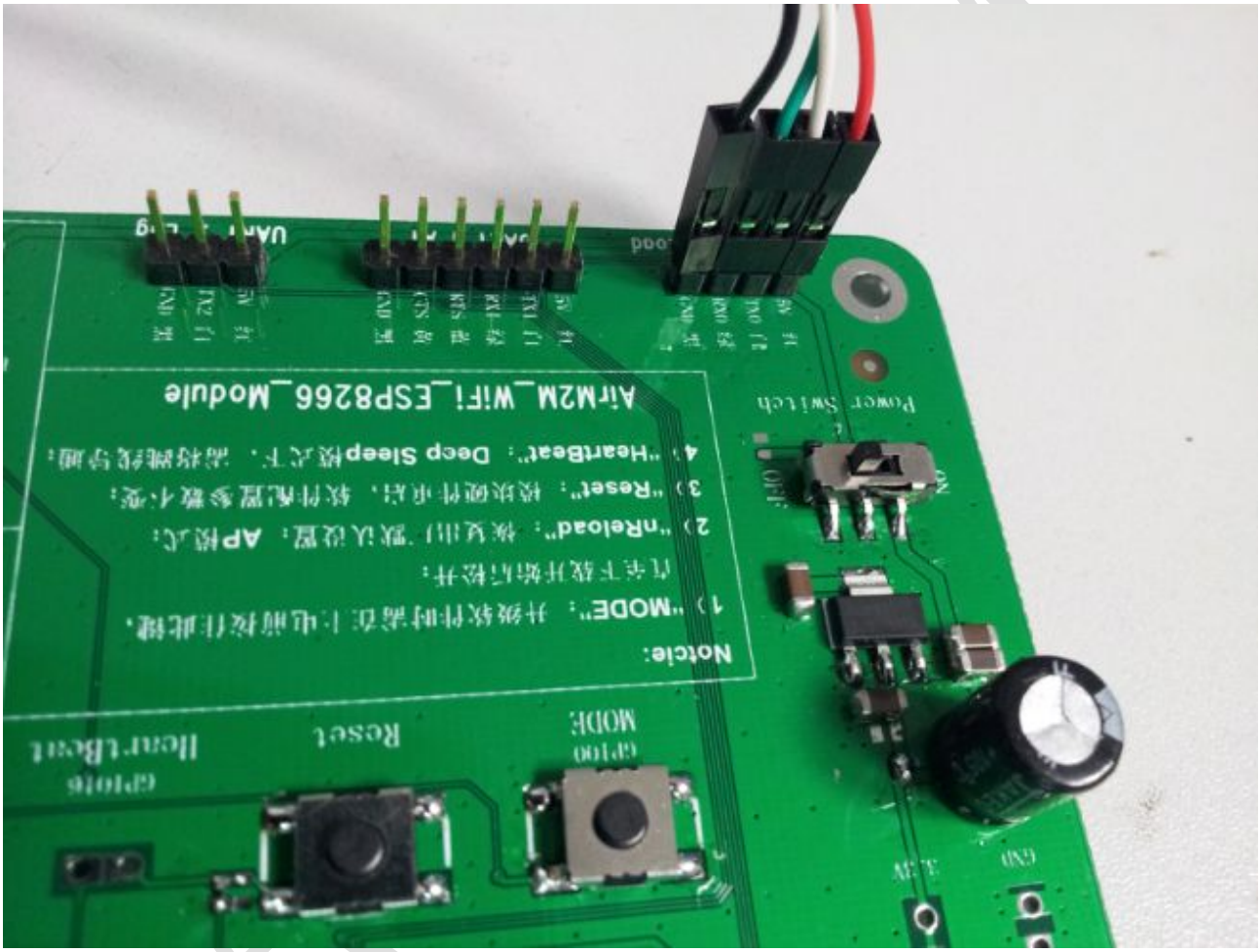
4) Button "reboot": module's hardware reboot, software configuration parameter won't change.

3.1. Module Firmware Download

The steps are:

1) Before firmware download, the A6501 module should be in shutdown state.
2) The firmware download is via UART0(PIN1=U0RXD,PIN2=U0TXD),so PC should be connected by UART-to-USB cable to A6501's UART0, and the cable's Tx should be connected to PIN1,the cable's Rx should be connected to PIN2.

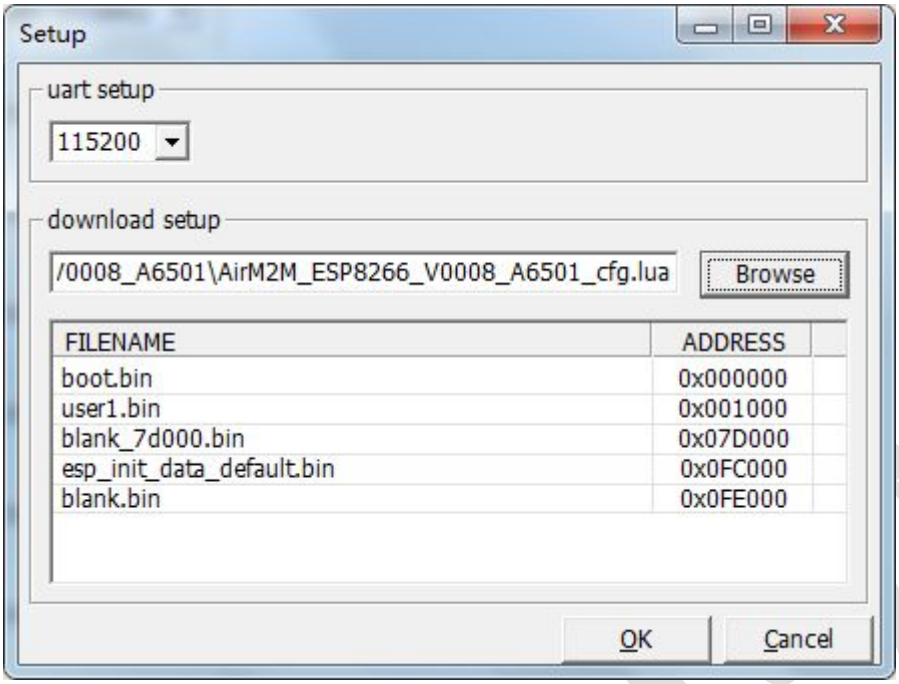
If the developer is using AirM2M's EVB board, the method for connection is indicated in the following picture:



3) Run the AirM2M_ESP8266_Firmware Download Tool on PC.

This tool can be run directly without installation. Open the tool and choose the port number for UART0.

4) Click the third toolbox button: "Setup" button  , choose the config file(_cfg.lua) in the new firmware directory and click "OK"



5) Click on the “download” button  , then pull down PIN 12(MODE) before powering on the module, then pull up MODE pin after the module goes into downloading mode.

If the developer is using AirM2M’s EVB board, the steps are:
Click on the “download” button on the PC tool -> press MODE button on the EVB -> power on the EVB->release the MODE button after the module comes into downloading mode.

6) When the prompt message: “FINISHED” appears, the downloading process is completed.

4. PC Configuration tool

4.1. Basic application methods

A650X supports using PC tools: AirM2M_ESP8266_CONFIG_TOOL to configure relative parameters for easy using.

Here we take how to set module to multilink transparent transmission mode(module working as STA+CLIENT) for example, introducing following configuration steps:

Step 1:
A650X wireless module is powered on, and in AP or AP+STATION mode(default is AP+STATION mode).

Step 2:
Put A650X module and the PC in the same LAN in order for the PC config tool to search out the module.
There are two ways:

a) Using computer's WIFI function to search out wireless module A650X (SSID of an AirM2M WiFi module as an AP is named as "AirM2M_MAC" e.g. AirM2M_97F502) and click connect.

b) PC and module join the same router. The SSID and password for the module to join the router can be provided to module by 2 means :

AT command (please refer:Start SMART LINK: AT+AMSL) or

SMARTLINK app(please refer:)

Step 3:

Open configuration tool AirM2M_WiFi_CONFIG_Tool. Please add trust if intercepted by PC security software.

Step 4:

Click Web Control -> SEARCH , the found module will be showed at the left-hand display column. If the module is not found, we can repeatedly click the button "SEARCH".

PC and module being in the same LAN is the precondition for the search function. Please refer to for details.

Step 5:

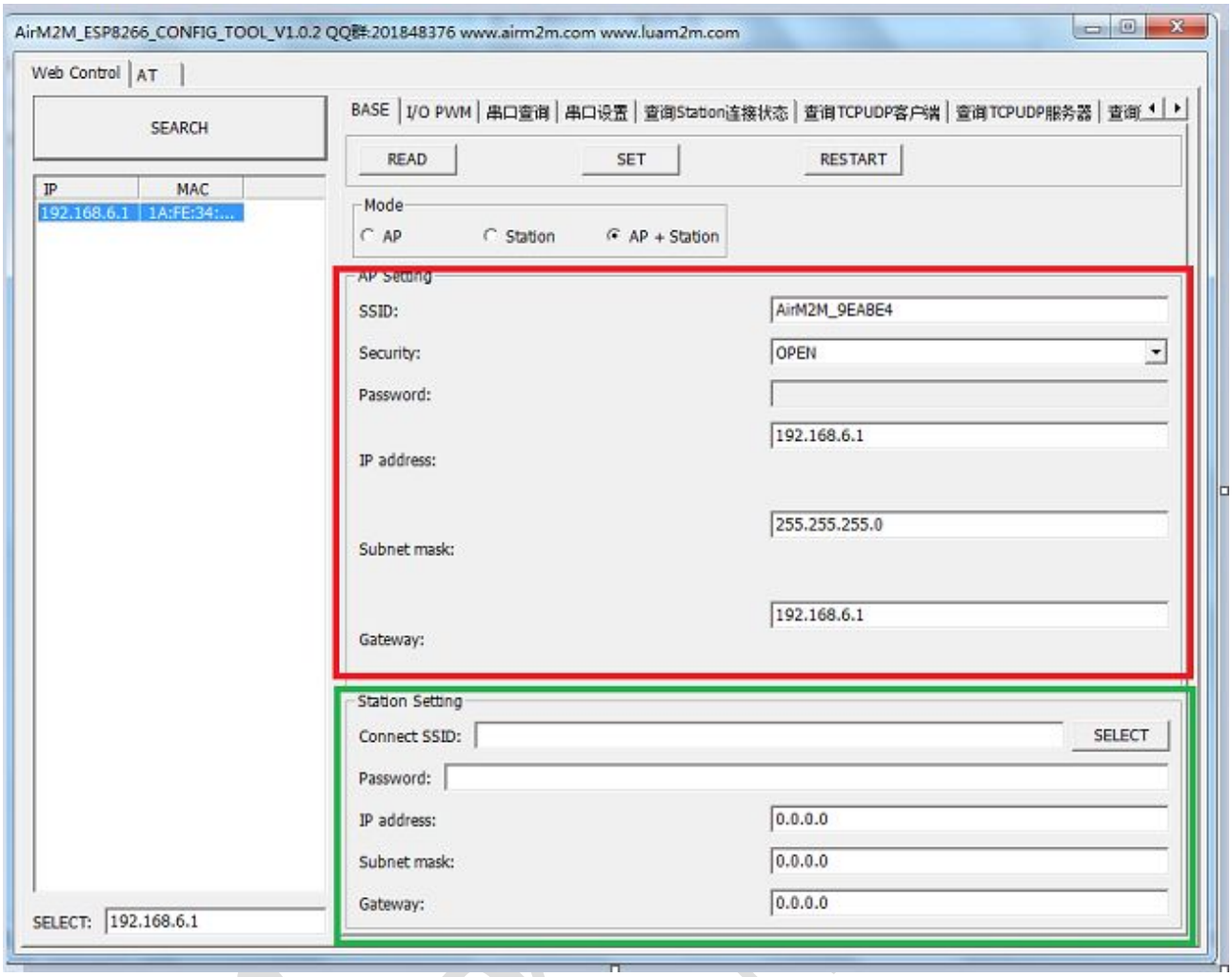
Click "BASE" tab ,and click "READ" button to get basic info of AirM2M WiFi module.

In the "BASE" tab, you can set the Wifi working mode and the related IP address.

Set the WiFi working mode to Station.

Setup the parameters in the green frame in the following picture(Ignore this part when module is in AP mode)

The parameters in the red frame are for AP configuration. Please ignore this part when module works as Station.



Note:

1) When working mode or IP address is changed, the module need to be restarted to make the parameters work;

2) If the IP address under the AP mode is changed , the developer need to do the search once more and select the module to use this tool after the restart of the module because the PC CONFIG Tool connects 192.168.4.1 without search by default .

Step 6:

Click “Set TCPUDP Client” tab to configure link. It supports 4 links at most at the same time.

You need set 4 parameters for every link:

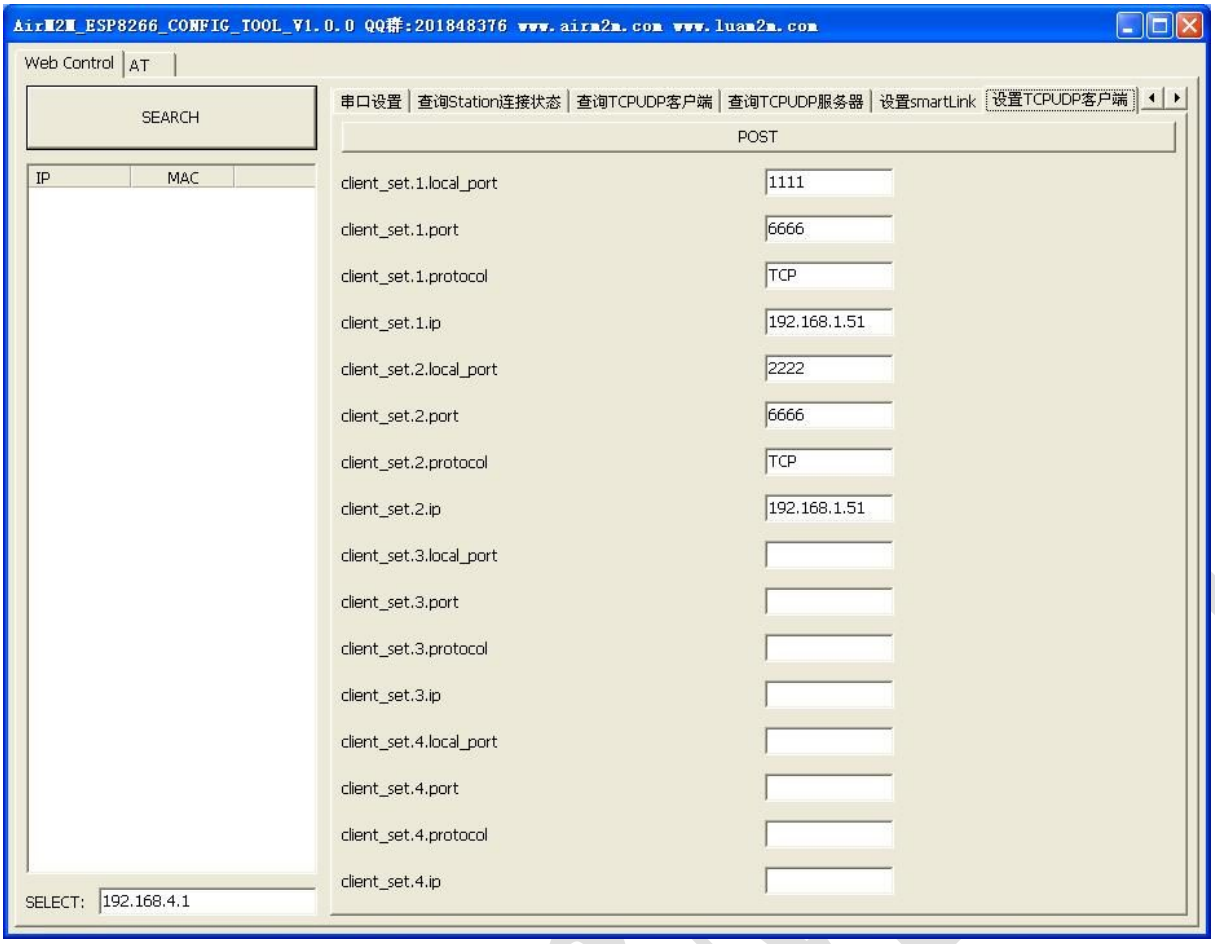
Local_port：local port

Port：server port

Protocol：TCP or UDP

Ip: server IP address

To configure links basing on need. Here take 2 for example and then click “POST”



Note: If user wants single link transparent transmission only, to configure one link at this step is enough.

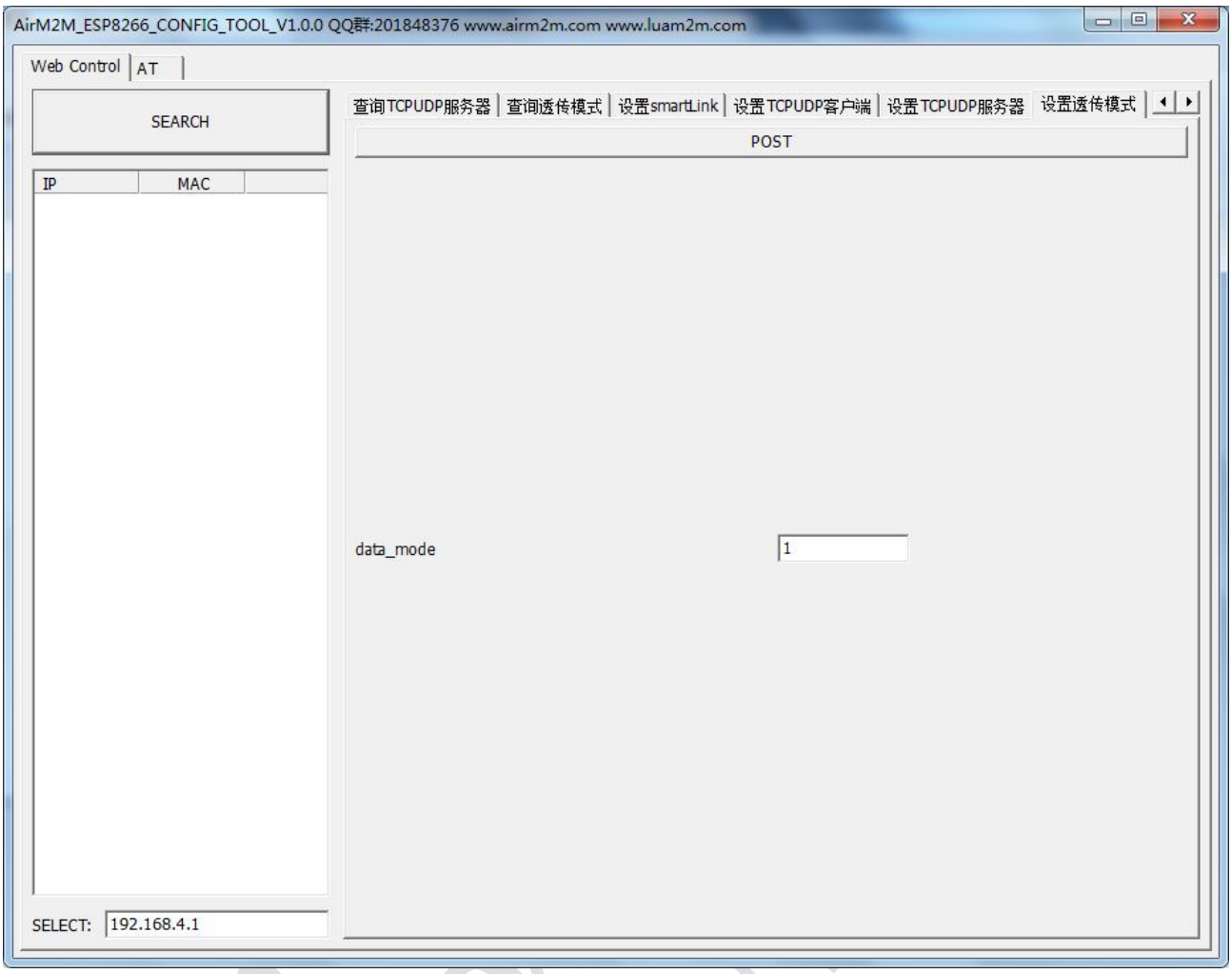
Step 7:

Click “set transparent transmission mode” option page to set transparent transmission mode.

Data_mode = 1 ,transparent transmission mode

Data_mode = 0 ,non-transparent transmission mode， that is AT command mode

Set data_mode=1 here， then POST

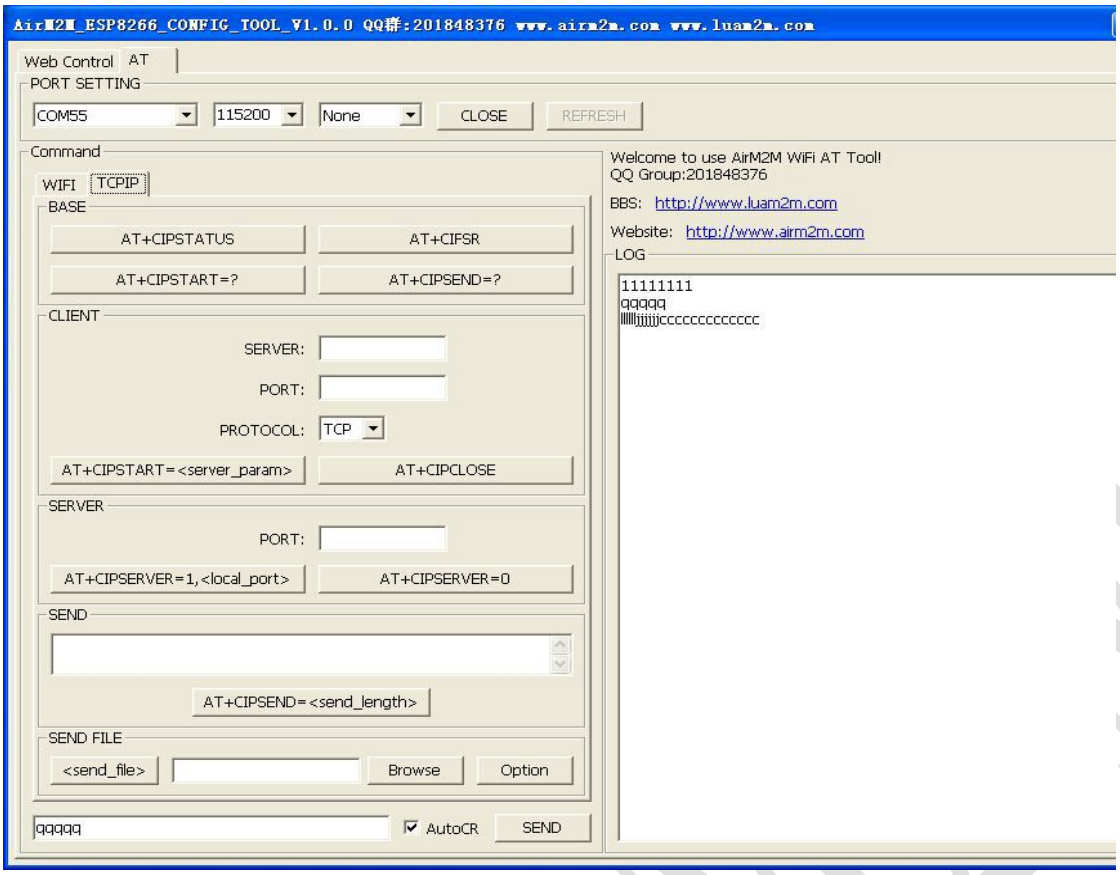


Step 8:

Reboot module. After rebooting, module will turn into transparent transmission state. Look from the server's side, there will be two links established.

Step 9:

Open AT tab, choose the port, put in data in the input box at the left-bottom corner and send. Look from the server's side, two links all receive data "qqqqq".



Step 10:

Quit transparent transmission mode and go back to AT command mode.

Click “set transparent transmission mode” set to set data_mode=0， POST, then reboot module.

Note:

There are 3 ways to quit transparent transmission mode:

1) +++ quit transparent transmission, but it will go back to transparent transmission mode after rebooting.
2) +++, then AT+CIPSCON=0, it will go to normal AT mode after rebooting module.
3) open AirM2M_ESP8266_CONFIG_TOOL->WEB CONTROL-> Set transparent transmission tab, set data_mode= 0 and then POST. Enter normal AT mode after rebooting module.

There are 2 ways to enter transparent transmission mode:

1) AT+CIPMODE=1,AT+CIPSERVER to configure server or AT+CIPSTART to configure client ,then AT+CIPSCON=1, reboot module and it will enter transparent transmission, and automatically set up connection according to last saved data (query-able by AT+CIPSCON)
2) open AirM2M_ESP8266_CONFIG_TOOL->WEB CONTROL-> Set transparent transmission tab, set data_mode= 1 and then POST. At Set TCPUDP client side/server tab to configure CLIENT/SERVER link, and then POST, reboot module to enter transparent transmission and automatically set up socket connection according to configuration.

4.2. Search AirM2M Module in the same LAN

In practical Wifi applications,such as Smart Home Systems,there is a scenario that most developers must consider: phone app or PC app need to search all the WIFI module/devices in the same home LAN in order to control them.

This function can be realized on APP in such method: **UDP broadcast.**

For APP, such parameters should be set:

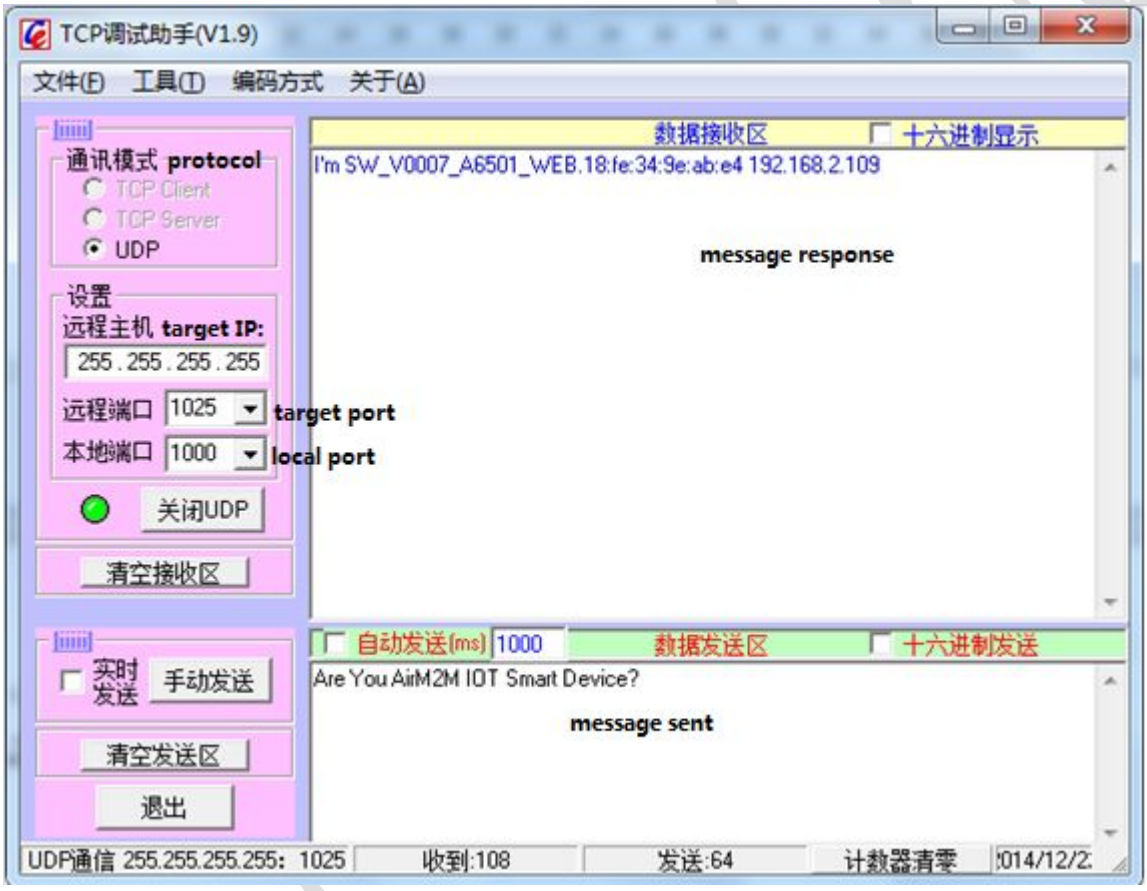
Broadcast IP address: 255.255.255.255

Broadcast port: 1025

Query packet from APP: Are You AirM2M IOT Smart Device?

Response from A6501:I'm FW_VER. MAC address IP address

Here is an PC APP example:



Note:

The search function in AirM2M_ESP8266_CONFIG_TOOL is implemented in this way too.

4.3. Smartlink Function

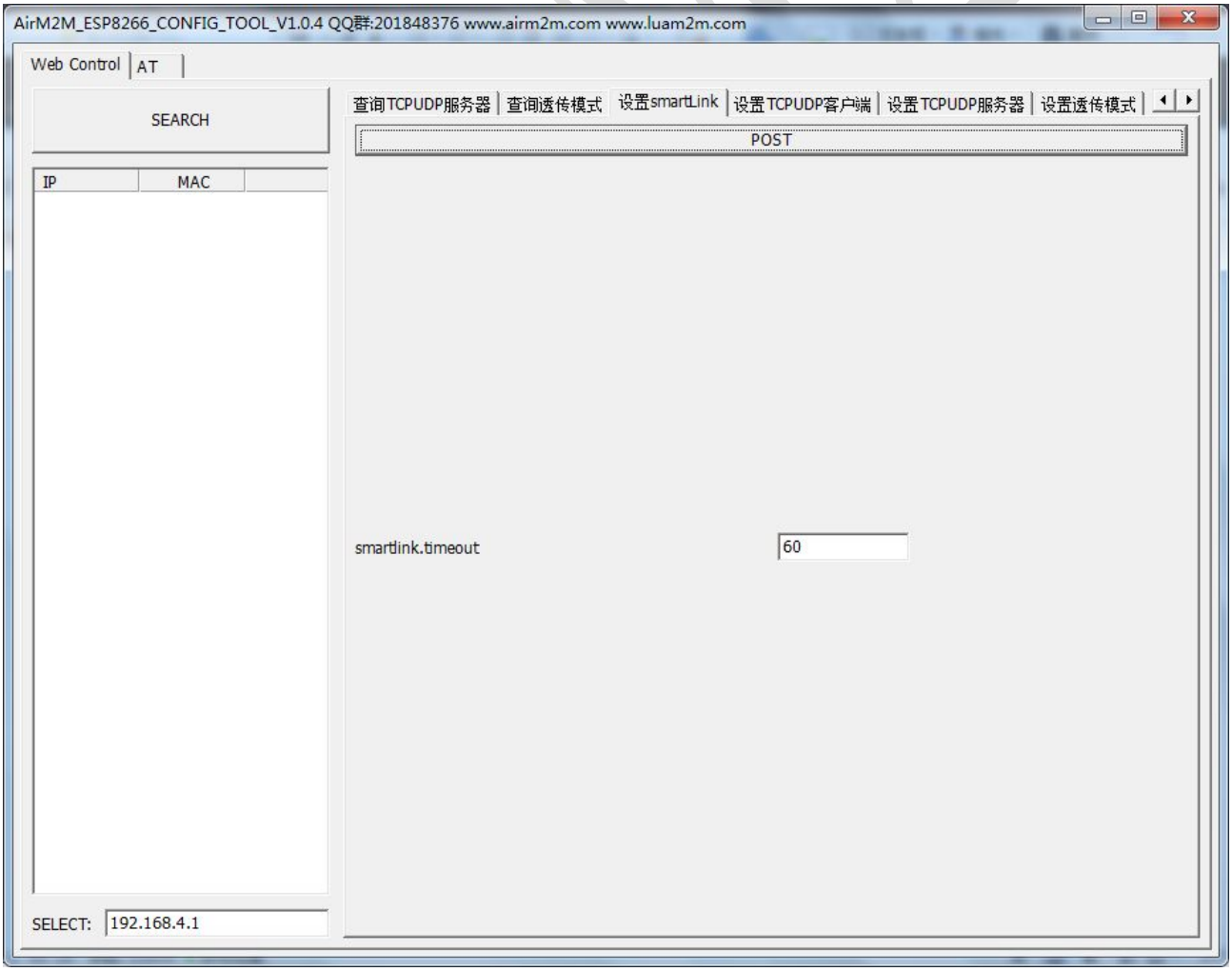
In the Wifi applications,module need to join APs. But modules have no keyboard or screen,it is not a easy thing for a module to join an AP .

Smart Link (sometimes called Smart Config), invented by TI(Texas Instruments), is a key to the solution.
The process is as follows:

1) The phone joins an router(=AP),
2) Open the the SmartLink phone App, input SSID and password of the router, then press Start, it will broadcast some packets which include ssid info,
3) Trigger the module to enter Smartlink process,the module will capture the broadcast packets and get the ssid info and then join the router according to the ssid info.

Note:
there are 3 ways to trigger the module's smartlink process for the AT command firmware version.

a) AT+AMSL=1. for details,please refer to:Start SMART LINK: AT+AMSL
b) Pull down nReload once
c) AirM2M_ESP8266_CONFIG_TOOL->"Set the SmartLink" tab, set the smartlink.timeout ≠ 0. reboot module , it will begin smartlink process repeatedly until it joins the router successfully.



This parameter describes the timeout of the smartlink packet capturing process,the unit is seconds. The default value is 0, which means no smartlink process at all.

For the SDK firmware version, the airm2m_smartlink_start API will trigger the Smartlink process once.(In SDK version , AT commands are not available.)

5. AT Instructions Introduction

5.1. Basic AT instruction

5.1.1 Reboot module：AT+RST

Syntax：

command type	grammar	response
execute command	AT+RST	OK

5.1.2 Set Uart baud rate：AT+IPR

Syntax：

command type	grammar	Response
set command	AT+IPR=<rate>	go back： OK
Read command	AT+IPR?	go back： +IPR: <rate> OK
test command	AT+IPR=?	go back： +IPR: (<rate> value list) OK
Items needing attention	Self-adaptive baud rate is supported by default. 1. The range of <rate> is: BIT_RATE_9600 BIT_RATE_19200 BIT_RATE_38400 BIT_RATE_57600 BIT_RATE_74880 BIT_RATE_115200 BIT_RATE_230400 BIT_RATE_256000 BIT_RATE_460800 BIT_RATE_921600 2. The baud rate is 921600 by default 3. Can be adaptive to baud rate that is no larger than the present one by training. For example,if the present baud rate is 115200,then any baud rate from among 9600、19200、	

	38400、57600、74880 can be achieved by sending several AT commands after power on via the UART . 4. The adaption is needed only when the baud rates between the module and the MCU are not the same. 5. The trained and adapted baud rate will be stored,which means the module's baud rate is the the last saved one before the rebooting. 6. The supported data bits: 5/6/7/8 The supported stop bits: 1/1.5/2 The supported parity: 0-none 1-odd 2-even The supported flow control: 0-none 1-hw control

5.1.3 Sleep and Wake-up Command：AT+AMSLEEP

Syntax：

command type	grammar	Response
set command	AT+AMSLEEP=<mode>,<sleep_time>	OK
Read command	AT+AMSLEEP?	+AMSLEEP:<mode>, <sleep_time> OK
test command	AT+AMSLEEP=?	+AMSLEEP:list of <mode>s, range of <sleep_time> OK

Parameter definition：

parameters	definition	value	value instruction
<mode>	sleep mode	2	deep sleep. only this mode is supported at present
<sleep_time>	time duration for sleep	0~4294	unit: seconds when the <sleep_time> is up, the module will wake up by itself, on the precondition that the PIN5(HEARTBEAT pin) is connected with RESET pin.

5.1.4 Cloud Firmware Update：AT+CIUPDATE

Syntax：

command type	grammar	Response
command	AT+CIUPDATE	+CIUPDATE: Update info OK

Examples：

cmd (->)/ Resp(<-)	AT sequences	Explanation
1） normal update		
->	AT+CIUPDATE	the precondition for firmware update of the module is that it join the router that can access the INTERNET.
<-	+CIUPDATE: CONNECT OK +CIUPDATE: A6501,V0008 +CIUPDATE: START UPGRADE OK	the module connects AirM2M Server automatically A6501 is the model,V0008 is the version of the new firmware
2） no need to update		
->	AT+CIUPDATE	
<-	+CIPUPDATE:CONNECT OK +CIPUPDATE:NO NEW VERSION AVAILABLE	the version of the running firmware is already the latest

5.1.5 Open log print：AT+OPENLOG

Syntax：

command type	grammar	Response
command	AT+OPENLOG	OK
Notes	the module firmware log is printed via nReload (PIN16) Pull down nReload once,start Smartlink process pull down nReload twice,start firmware update process pull down nReload for a long time(>3s), to restore factory settings Input AT+OPENLOG,the functions mentioned above will be disabled. Reboot the module to restore.	

5.1.6 Enable Command Echo：ATE

The ATE command determines whether or not the TA echoes characters received from TE during command state.

Syntax：

command type	grammar	Response
command	ATE<value>	OK

Parameter definition：

parameters	definition	value	value instruction
<value>		0	echo mode off

		1	echo mode on,default value

5.1.7 Set Flow Control：AT+IFC

Syntax：

command type	grammar	Response
Set command	AT+IFC=<mode>	OK

Parameter definition：

parameters	definition	value	value instruction
<mode>		0	flow control OFF
		1	HARDWARE

5.2. WiFi function AT instruction

5.2.1 Select WiFi working mode：AT+CWMODE

Syntax：

command type	grammar	Responseuction
Set command	AT+CWMODE = <mode>	OK
		(AT+RST)Command will be in valid after reboot
Read command	AT+CWMODE?	+CWMODE:<mode> OK
		What mode is currently in?
test command	AT+CWMODE = ?	+CWMODE:(<mode> value list) OK
		What mode it support currently?

Parameter definition：

parameters	definition	value	value instruction
<mode>	WiFi application mode	1	Station mode
		2	AP mode
		3	AP+Station mode

5.2.2 List current available access points: AT+CWLAP

Syntax：

command type	grammar	Response
execute command	AT+CWLAP	+CWLAP: <ecn>,<ssid>,<rssi>[,<mode>] OK
		this command will go back to AP list

Parameter definition：

parameter	definition	value	value instruction
<ecn>	encryption way	0	OPEN
		1	WEP
		2	WPA_PSK
		3	WPA2_PSK
		4	WPA_WPA2_PSK
<ssid>	access point name		character string parameter
<rssi>	signal strength		
<mode>	connect mode	0	manual connect
		1	automatic connection

5.2.3 Add access point:AT+CWJAP

Syntax：

command type	Grammar	Response
Set command	AT+CWJAP=<ssid>,<pwd>	OK or ERROR
		Successfully join AP, OK. If not, ERROR
Read command	AT+CWJAP?	+CWJAP:<ssid> OK
		go back to current choice AP

Parameter definition：

parameter	definition	value	value instruction
<ssid>	access point name		character string
<pwd>	password		character string, the longest is 64 byte, ASCII coding

5.2.4 Quit access point: AT+CWQAP

Syntax：

command type	grammar	Response
execute command	AT+CWQAP	OK
		means exit this AP successfully
test command	AT+CWQAP=?	OK
		check whether this command is supported

5.2.5 Set parameter in AP mode: AT+CWSAP

Syntax：

command type	grammar	Response
set command	AT+CWSAP=<ssid>,<pwd>,<chl>, <ecn>	OK
		successfully set parameters
Read command	AT+CWSAP?	OK
		check current AP parameter

parameter definition：

parameter	definition	value	value instruction
<ecn>	encryption way	0	OPEN
		1	WEP
		2	WPA_PSK
		3	WPA2_PSK
		4	WPA_WPA2_PSK
<ssid>	access point name		character string parameter
<pwd>	password		character string, the longest is 64 byte, ASCII coding
<chl>	channel number		

5.2.6 Inquire MAC address: AT+AMMAC

Syntax：

command type	grammar	Response
execute command	AT+AMMAC	If CWMODE set as 1（STA mode）： STATION_MAC: xx:xx:xx:xx:xx:xx OK
		If CWMODE set as 2（AP mode）：

		AP_MAC: xx:xx:xx:xx:xx:xx OK
		If CWMODE set as 3（STA+AP mode）： STATION_MAC: xx:xx:xx:xx:xx:xx AP_MAC: xx:xx:xx:xx:xx:xx OK
		Successfully set up parameters

Examples：

cmd（→）/Rep（←）	example	explanation and instruction
→	AT+CWMODE?	Inquire WIFI level working mode
←	+CWMODE::3 OK	3：AP+STA mode
→	AT+AMMAC	Inquire AC address
←	STATION_MAC: 18:fe:34:9e:ab:e4 AP_MAC: 1a:fe:34:9e:ab:e4 OK	
→	AT+CWMODE=2	Set working mode to 2，AP mode
←	OK	
→	AT+AMMAC	Inquire MAC adress
←	AP_MAC: 1a:fe:34:9e:ab:e4 OK	

5.2.7 Query the signal strength of the AP: AT+CAPR

Syntax：

command type	grammar	Response
execution command	AT+CAPR	+CAPR:<ssid>,<rssi> OK

parameter definition：

parameter	definition	value	Value instruction
<ssid>	AP hot spot name		the name of the joined AP by the module
<rssi>	AP signal strength		

Examples：

cmd（→）/Rep（←）	example	explanation and instruction
→	AT+CAPR	query the strength of the AP joined by the module currently
←	+CAPR:"AIRM2M", -65 OK	AP's name is AIRM2M,the rssi is -65

5.2.8 Start SMART LINK: AT+AMSL

Syntax：

Command type	grammar	Response
Set command	AT+AMSL=1	+AMSL:<ssid>,<password> OK
		successfully configured SMART LINK

parameter definition：

parameter	definition	value	Value instruction
<ssid>	AP hot spot name		
<password>	AP hot spot password		

Examples：

cmd (->)/ Resp(<-)	example	Explanation and instruction
		• Turn on cell phone's WIFI function and link to a router (name it A for easy understanding). • Turn on SMARTLINK Android App that released by our company. Input name and password of a router that we want module to link to(named B). And click SMARTLINK button below. Keep it running until configuration finish. Note: A and B can be the same router. That means that cell phone can directly link to router B that we want module to join, and configure router B in APP.
→	AT+CWMODE?	Inquire WIFI level working mode

←	+CWMODE:1 OK	Mode 1：STA mode Mode 3：STA+AP mode Mode 1 and 3，both can use SMART LINK But mode 2(pure AP mode)cannot configure SMARTLINK.
→	AT+AMSL=1	Start SMART LINK configuration.
←	+AMSL: yh_taih,qin02666 OK	It means configuration is successful Ssid = yh_taih Password = qin02666
→	AT+CWJAP?	Inquire which hotspot module joins up currently.
←	+CWJAP:"yh_taih" OK	means successfully joined up yh_taih
		Click STOP，turn off APP

5.3. TCPIP AT command

5.3.1 set up TCP/UDP connection ： AT+CIPSTART

Syntax：

command type	grammar	Response
set command	when single way connection (+CIPMUX=0)： AT+CIPSTART=<type>,<addr>,<port>,[local_port] When multichannel connection: AT+CIPSTART=<id>,<type>,<addr>,<port>,[local_port] And local port is optional parameters	if format is right： OK otherwise： +CME ERROR: invalid input value successfully connect ： CONNECT OK (CPIMUX=0) <id>, CONNECT OK (CIPMUX=1) If connection already exist： ALREADY CONNECT connection fail ： CONNECT FAIL (CIPMUX=0) <id>, CONNECT FAIL (CIPMUX=1)

parameter definition：

parameter	definition	value	value instruction
<id>	Link No.	0~3	if AT+CIPMUX=0,the value of <id> is fixed to 0
<type>	connection type	“TCP”/”UDP”	
<addr>	Remote server IP address		character string IP address or domain name
<port>	remote service port number		
[local_port]	local port		number, optional parameters

5.3.2 Get TCP/UDP connection mode ： AT+CIPSTATUS

Syntax：

command type	grammar	Response

execute command	AT+CIPSTATUS	If it is single connection (AT+CIPMUX=0): OK STATE: <state> C:<cid>, <TCP/UDP>, <Local IP address>,<Local port>,<Remote IP address>, <Remote port>, <client state> Note: If it is single connection, it uses the following way for connection: AT+CIPSTART=<type>,<addr>,<port>,[local_port] way, and one link can be set up only. The occupied <cid> = 0
		If it is mult connection (AT+CIPMUX=1): OK STATE:<state> S: <sid>,<port>,<server state> C:<cid>, <TCP/UDP>, <Local IP address>,<Local port>,<Remote IP address>, <Remote port>, <client state>
test command	AT+CIPSTATUS=?	OK

parameter definition：

parameter	definition	value	value instruction
<state>	single connection state	IP INITIAL	initialization
		IP STATUS	obtain local IP status
		TCP CONNECTING/UDP CONNECTING	TCP connecting/UDP port registering
		CONNECT OK	successfully connect setup
		TCP CLOSING/UDP CLOSING	Shutting down TCP connection, and logging out UDP port.
<sid>	server ID	0~1	value 0 and 1
<server state>	server status	OPENING	turning on
		LISTENING	monitoring
		CLOSING	turning off
<cid>	customer's ID	0~3	value as 0,1,2,3
<Local IP address>	local IP		
<Local port>	local port		
<Remote IP address>	remote IP		
<Remote port>	remote port		
<client state>	client status	INITIAL	turn off state
		CONNECTING	connecting
		CONNECTED	connected

Examples：

cmd(->)/ Resp(<-)	example	Explanation and instruction
→	AT+CWMODE?	Inquire WIFI level working mode
←	+CWMODE:3 OK	3：AP+STA mode
→	AT+CIPMUX?	
←	+CIPMUX:0 OK	Single connection mode
→	AT+CIPSTATUS	
←	OK STATE:IP STATUS C:0,"","",,"",,"INITIAL" C:1,"","",,"",,"INITIAL" C:2,"","",,"",,"INITIAL" C:3,"","",,"",,"INITIAL"	
→	AT+CIPSTART="TCP","192.168.1.51",6800	Set up one connection as client
←	OK CONNECT OK	
→	AT+CIPSTATUS	
←	OK STATE:CONNECT OK C:0,"TCP",”192.168.1.52”,38330,"192.168.1.51",6800,"CONNECTED" C:1,"","",,””,,"INITIAL" C:2,"","",,””,,"INITIAL" C:3,"","",,””,,"INITIAL"	
→	AT+CIPCLOSE	close the connection
←	CLOSE OK	
→	AT+CIPMUX=1	Must close all connection before configure mult- connection
←	OK	
→	AT+CIPSERVER=1,7777	
←	OK	
←(URC)	0,CONNECT OK	module accepts a socket link
→	AT+CIPSTATUS	
←	OK STATE:IP STATUS S:0,7777,LISTENING C:0,"TCP","192.168.4.1",7777,"192.168.4.101",59319,"CONNECTED" C:1,"","",,"",,"INITIAL" C:2,"","",,"",,"INITIAL"	

	C:3,"","",,"",,"INITIALC:2,"","",,"",,"INITIAL"	

5.3.3 Start multi-connection： AT+CIPMUX

Syntax：

command type	grammar	Response
set command	AT+CIPMUX=<mode>	The response is : OK Note:Start multi-connection successfully
		If it is in multi- connection mode,the response is: +CME ERROR: The socket is already connected Note:That is when there already is a connection for module, <mode> value can not be modified. Only after all connections are closed by AT+CIPCLOSE, <mode> can be modified.
Read command	AT+CIPMUX?	+CIPMUX:<mode> OK

parameter definition：

parameter	definition	value	value instruction
<mode>	whether to boot multi-connection mode	0	Configure to single connection mode
		1	Configure to multi- connection mode

5.3.4 Send data： AT+CIPSEND

Grammatical rues：

command type	Grammar	Response	
set command	single connection(+CIPMUX=0)： AT+CIPSEND=<length>	response	When module receives command it will feed back ”>” first, then start to receive serial port data. Data will be sent automatically when its length reaches <length>. If connection is not setup,the module responds: ERROR If successfully sent data, the module responds:SEND OK. If parameters error：
	multi connection(+CIPMUX=1)： AT+CIPSEND=<id>,<length>		

			+CME ERROR: invalid input value
		description	send data of assigned length
test command	AT+CIPSEND=?	response	single connection(AT+CIPMUX=0): +CIPSEND: <length> OK
			multi- connection(AT+CIPMUX=1): +CIPSEND: <0-3>,<length> OK
execute command	AT+CIPSEND	description	AT+CIPMODE=1 and as a client, this command will make the module enter transparent transmission mode(hardware flow control is recommended in this circumstances, otherwise data mayl be lost when there are mass data.) When module receives command it will send back ”>” first, then start to send data h received via serial port.

Parameter definition：

parameter	definition	value	value instruction
<length>	data length		unit: bytes
<id>	link No.	0~3	connection id

5.3.5 Set the TCP socket window size：AT+CIPWND

command type	grammar	Response
set command	AT+CIPWND=<linkid>,<tcp_window_size>	OK
execute command	AT+CIPWND?	+CIPWND:<linkid>,<tcp_window_size> OK
test command	AT+CIPWND=?	+CIPWND:(range of <linkid>), (range of <tcp_window_size>)

		OK

Parameter definition：

parameter	definition	value	value instruction
<linkid>	connection id	0~3	
<tcp_window_size>	TCP window size	0~1024	unit: bytes

Examples:

cmd(->)/ Resp(<-)	example	Explanation and instruction
→	AT+CIPWND=?	
←	+CIPWND:(0-4),(0-1024) OK	
→	AT+CIPSTART="TCP","192.168.1.51",6800	建立一个单链接
←	OK CONNECT OK	连接上了
→	AT+CIPWND?	
←	+CIPWND: C:0,5840 C:1, C:2, C:3, OK	
→	AT+CIPWND=0,100	
←	+CIPWND: C:0,100 C:1, C:2, C:3, OK	

5.3.6 Set the frame size for transparent transmission： AT+CIPCCFG

Set the frame size for automatic sending in transparent data transmission mode.

Syntax:

command type	grammar	Response

set command	AT+CIPCCFG=<wait_time>,<frame_size>	OK
execute command	AT+CIPCCFG?	+CIPCCFG:<wait_time>,<frame_size> OK
test command	AT+CIPCCFG=?	+CIPCCFG:range of <wait_time>,range of <frame_size> OK

Parameter definition：

parameter	definition	value	value instruction
<wait_time>	waiting time	1~65535	unit:ms , default value =100 when the waiting time exceeds the set value ,the data are sent automatically
<frame_size>	frame size	1~65535	unit:bytes, default value = 2048 when the frame size exceeds the set value,the data are sent automatically

5.3.7 Close TCP/UDP connection：AT+CIPCLOSE

Syntax：

command type	grammar	Response
set command	Single connection AT+CIPCLOSE=<id>	CLOSE OK
	Multi-connection AT+CIPCLOSE=<n>[,<id>]	<n>,CLOSE OK
execution command	AT+CIPCLOSE	If shut down successfully： CLOSE OK If shut down fail： ERROR
test command	AT+CIPCLOSE=?	OK
Items need attention	● Execution command is effective to single connection only, it will respond ERROR in multi-connection. ● Execution command AT+CIPCLOSE is only available when module is in TCP/UDP CONNECTING or CONNECT OK status, in otherwise it will return ERROR. ● Status after shutting down is IP CLOSE in single connection mode.	

parameter definition：

parameters	definition	value	value instruction
<id>	shut down mode	0	slow shutdown(default value)
		1	quit shutdown
<n>	Link No.	0~3	integer type, connection serial number

5.3.8 Obtain local IP address：AT+CIFSR

Set command can set the module's IP address.

Read command can read the module's current IP addresses under AP and/or STA modes.

Syntax：

command type	grammar	Responses	
Set command	AT+CIFSR=<mode>,<ip_addr>,<netmask>,<gateway>	success	OK
		fail	+CME ERROR: invalid input value
Read command	AT+CIFSR	CWMODE=1	+CIFSR:<sta_IP address> OK
		CWMODE=2	+CIFSR:<ap_IP address> OK
		CWMODE=3	+CIFSR:<sta_IP address> +CIFSR:<ap_IP address> OK
		fail to get IP	ERROR
test command	AT+CIFSR=?	success	OK

parameter definition：

parameter	definition	value	value instruction
<mode>	WIFI mode	1	Station
		2	AP
<ip_addr>	Set IP address		格式为 xx.xx.xx.xx
<netmask>	net mask		格式为 xx.xx.xx.xx
<gateway>	gateway		格式为 xx.xx.xx.xx
<sta_IP address>	IP address in STA mode		格式为 xx.xx.xx.xx
<ap_IP address>	IP address in AP mode		格式为 xx.xx.xx.xx

5.3.9 Configure module as server：AT+CIPSERVER

Need execute AT+CIPMUX=1 first to turn on mux mode. To monitor 2 ports at most at the same time, including TCP and UDP.

Syntax：

command type	grammar	Responses	
Execute command	AT+CIPSERVER=<mode>[,<port>]	success	OK
		If there is a client to connect the module(as a server)	<linkid>,CONNECT OK

parameter definition：

parameter	definition	value	value instruction
<mode>	whether turn on server mode	0	turn off server mode Note: Need reboot if use AT+CIPSERVER=0 to shut down server.
		1	turn on server mode
<port>	port number		default value is 333

5.3.10 Receives data from server：+IPD and+RECEIVE

The URC（Unsolicited Result Code） is sent by module to serial port. When module receives network data it sends data which uses +IPD and RECEIVE as header to serial port.

Syntax：

command type	grammar	Response and instruction
non-transparent transmission	single connect (+CIPMUX=0): +IPD,<len>:<data>	received data will take +IPD as beginning.
	multi- connect (+CIPMUX=1): +RECEIVE,<id>,<len>:<data>	<data> is received data, there is one \r\n between it and +RECEIVE, <n>,<length>: Note: Colon is an English colon.
transparent transmission	<data>	There is no data head, data come up directly when it is transparent transmission. **Note:** **Ways for transparent transmission for**

		multi-connections are: Every single data sent to module serial port will be automatically sent to all connections under CONNECTED state by module. Users can guarantee data's pertinence by application layer's protocol.

Parameter definition：

parameter	definition	value	value instruction
<id>	Link No.	0~3	connection serial number
<len>	<data> length		unit: bytes Note: The length is <data>'s length, data head not included.
<data>	burst		

Example of data receiving(take receiving "TEST123" for example)：

	Non-transparent transmission	Transparent transmission
single link	+IPD,7:TEST123	TEST123
multi-link	+RECEIVE,1,7: TEST123	TEST123(at every CONNECTED connection）

5.3.11 Select TCPIP application mode ： AT+CIPMODE

Syntax：

command type	grammar	backward
set command	AT+CIPMODE=<mode>	OK
Read command	AT+CIPMODE?	+CIPMODE: <mode> OK
test command	AT+CIPMODE=?	+CIPMODE: (0-NORMAL MODE,1-TRANSPARENT MODE) OK

parameter definition：

parameter	definition	value	value instruction
<mode>	TCPIP application mode	0	Non-transparent transmission mode, default mode.
		1	transparent transmission mode

5.3.12 Save transparent transmission configuration：AT+CIPSCON

Syntax：

command type	grammar	backward

set command	AT+CIPSCON=<action>	OK
Read command	AT+CIPSCON	C:<id>,<protocol>,<ip>,<remotePort>,<localPort> S:<id>,<localPort>,<timeout> DATA_MODE:<mode> OK
test command	AT+CIPSCON=?	+CIPSCON:(0,1) OK

parameter definition：

parameter	definition	value	value instruction
<action>	Save/delete transparent transmission parameter	1	Save transparent transmission parameter Module enters transparent transmission after booting and : 1) Join the AP according to AT+CWJAP 2) Set up connection according to data of C: and monitor port according to data of S:.
		0	delete transparent transmission parameter. Data of S and C is deleted. Module quits transparent transmission mode at the same time
<id>	connection id		
<protocol>	TCPIP protocol	TCP	
		UDP	
<ip>	Ip address		x.x.x.x
<remotePort>	remote port		means server port
<localPort>	local port		To C, that is module's local CLIENTport To S，that is module's loca SERVER port
<timeout>	Server link time-out		Please refer to AT+CIPSTO. Default value is 180
<mode>	Transparent transmission mode	0	Common AT mode
		1	Transparent transmission mode

Examples：

cmd(->) /resp(<-)	example	Explanation and instruction
→	AT+CWMODE?	
←	+CWMODE:1 OK	
→	AT+CIPMODE=1	
←	OK	
→	AT+CIPMUX=1	
←	OK	
→	AT+CWJAP="ChinaNet-hXug","5qsqruiv"	
←	OK	

→	AT+CIPSTART=0,"TCP","192.168.1.51",6800	
←	OK 0,CONNECT OK	
→	AT+CIPSCON=1	
←	OK	
→	AT+CIPSCON	
←	C:0,"TCP","192.168.1.51",6800,1793 C:1,"","", , C:2,"","", , C:3,"","", , S:0, , S:1, , DATA_MODE:1 OK	
	AT+CIPSTATUS	
	OK STATE:IP STATUS C:0,"TCP","192.168.1.51",6800,"CONNECTED" C:1,"","",,"INITIAL" C:2,"","",,"INITIAL" C:3,"","",,"INITIAL"	
Then reboot module. Module will enter transparent transmission mode after reboot and automatically set up connection of 0,"TCP","192.168.1.51",6800,"CONNECTED"		

5.3.13 Time-out for server disconnection：AT+CIPSTO

command type	grammar	Response
set command	AT+CIPSTO=<server timeout >	OK
Read command	AT+CIPSTO?	+ CIPSTO:<server timeout> OK

parameter definition：

parameter	definition	value	value
<server timeout >	time-out for server automatically disconnected.	0~4294967295	unit is 1s Using this command to set up time-out time, server will disconnect when time is up. 0 is default value. Means never to take initiative to disconnect. Default value is 180s.

Examples：

cmd(->) /resp(<-)	example	Explanation and instruction
→	AT+CIPSTO=0,120	set server timeout = 120 seconds
←	OK	
→	AT+CIPSTO?	query the settings
←	S:0,120 S:1,180	

5.3.14 Select non-transparent transmission data sending mode : AT+CIPQSEND

Under non-transparent transmission data sending mode, there are two sending modes quick sending and slow sending also, using +CIPQSEND to set up.

Syntax:

command type	grammar	Backward
set command	AT+CIPQSEND=<n>	OK
Read command	AT+CIPQSEND?	+CIPQSEND: <n> OK
test command	AT+CIPQSEND=?	+CIPQSEND: (0,1) OK

parameter definition:

parameter	definition	value	value instruction
<n>	non-transparent transmission mode	0	Common mode, also called slow sending mode—when server receives TCP data, module goes back to SEND OK.
		1	Quick sending mode—when data sent to module, module single link: DATA ACCEP:<length> Multi-link: DATA ACCEPT:<id>,<length> but not SEND OK

5.3.15 Set the reconnection times on a TCP link : AT+CIPRCON

Set command determines the re-connection times after an abnormal link disconnection happens on a TCP link.

In either data sending modes ,normal AT mode or transparent transmission mode, on a TCP link, a packet of data input will do the service of discerning the link connection state:CONNECTED or otherwise.

Syntax:

command type	grammar	Backward

set command	AT+CIPRCON=<times>	OK
Read command	AT+CIPRCON?	+CIPRCON:<times> OK
Test command	AT+CIPRCON=?	+CIPRCON:range of <times> OK

parameter definition：

parameter	definition	value	value
<times >	re-connection times after an abnormal TCP link disconnection	0~65535	Unit:times,default value = 3 . 0= never reconnect 65535 = always reconnect

5.3.16 Quit transparent transmission mode: + + +

There are 3 ways to quit transparent transmission mode:

4) +++ quit transparent transmission, but it will go back to transparent transmission mode after rebooting.
5) +++, then AT+CIPSCON=0, it will go to normal AT mode after rebooting module.
6) open AirM2M_ESP8266_CONFIG_TOOL->WEB CONTROL-> Set transparent transmission tab, set data_mode= 0 and then POST. Enter normal AT mode after rebooting module.

There are 2 ways to enter transparent transmission mode:

3) AT+CIPMODE=1,AT+CIPSERVER to configure server or AT+CIPSTART to configure client ,then AT+CIPSCON=1, reboot module and it will enter transparent transmission, and automatically set up connection according to last saved data (query-able by AT+CIPSCON)
4) open AirM2M_ESP8266_CONFIG_TOOL->WEB CONTROL-> Set transparent transmission tab, set data_mode= 1 and then POST. At Set TCPUDP client side/server tab to configure CLIENT/SERVER link, and then POST, reboot module to enter transparent transmission and automatically set up socket connection according to configuration.

Syntax：

Command type	grammar	backward
execute command	+++	Attention： +++ no other character at front or back

5.3.17 Application examples

Module can work as STATION on WIFI level, also can work as AP, and AP+ STATION. Module can work as SERVER or CLIENT on transmission level. While on application level, it can work in non-transparent data transmission (normal AT mode) and transparent data transmission mode as well.

Examples：

cmd(→) /resp(←)	example	explanation and demonstration
1）Module works in STATION mode（mode=1）： The parts in light blue grids are the common premise steps in this mode：		
→	AT+CWMODE?	Step1: check WIFI level working mode
←	+CWMODE::3 OK	Default mode is 3: AP+STA mode
→	AT+CWMODE=1	Step2: Set as Station mode
←	OK	
→	AT+RST	Step3: Need reboot after setting operation mode.
←	OK	**Note: Step2 and Step 3 can be ignored if after booting AT+CWMODE? is already Step 1.**
→	AT+CWLAP	Step 4: check current available access point
←	+CWLAP:(3,"yh_taihdhad",-45,"54:e6:fc:52:36:e6",9) +CWLAP:(4,"lemon",-86,"6c:e8:73:69:4b:76",10) +CWLAP:(4,"Xiaomi_yu",-85,"8c:be:be:2c:46:ea",11) OK	result
→	AT+CWJAP="yh_taihdhad","UUU02339"	Step 5: join one AP
←	OK	**Note: After finishing this step, it will automatically save the WIFI working mode and the AP information.**
1.1）Module works as CLIENT, single connection, non-transparent transmission：		
→	AT+CWJAP?	Check current joined AP.
←	+CWJAP:"yh_taihdhad" OK	
→	AT+CIFSR	Check current module IP address
←	172.168.1.114	
→	AT+CIPSTART="TCP","172.168.1.119",4321	Set up one TCP link
←	OK	
←	CONNECT OK	Means connection success.
→	AT+CIPSEND=5	Set sending length to 5, module will automatically send data after data up to 5 ones.
→	>hello	After > comes up, send 5 characters hello to serial port, module will automatically send data.
←	SEND OK	means sending is successful
←	+IPD,14:hello,thisispc	Module receives server's data and send to

		serial port directly.
→	AT+CIPCLOSE	Turn off connection
←	OK	
1.2）Module works as CLIENT, single connection, data transparent transmission：		
→	AT+CIPMODE=1	set module to transparent transmission mode
←	OK	
→	AT+CIPSTART="TCP","172.168.1.119",4321	connect to one server
←	OK	
←	CONNECT OK	means connection is successful
→	AT+CIPSCON=1	Save parameter(link parameter and transparent transmission mode)
←	OK	
→	AT+CIPSEND	Enter transparent transmission. Enter transparent transmission directly without reboot.
←	>	After enter AT+CIPSEND, it will send back >, then MCU can send data to module from serial port, module can automatically send out data.
←	SSSSAAAAAAAA	After server sending “SSSSAAAAAAAA”, module will display directly. Note: No data head when received by transparent transmission.
→	Reboot module	Module will enter transparent transmission directly after reboot and set up connection according to "TCP","172.168.1.119",4321.
→	+++	quit transparent transmission state Note: there should be no other characters at front or behind +++.
→	AT+CIPSCON	Inquire transparent transmission parameter and state
←	C:0,"TCP","172.168.1.119",4321,1793 C:1,"","", , C:2,"","", , C:3,"","", , S:0, , S:1, , DATA_MODE:1 OK	Inquire result is status and parameter of the last time when AT+CIPSCON=1.
→	AT+CIPSTATUS	
←	OK STATE:CONNECT OK C:0,"TCP",”172.168.1.121”,35330,"172.168.1.119",4321,"CONNECTED" C:1,"","",,””,,"INITIAL" C:2,"","",,””,,"INITIAL" C:3,"","",,””,,"INITIAL"	
→	Reboot module. Module will re-enter transparent transmission after reboot and set up one connection automatically. Connection parameter is the same as saved last time.	

1.3）Module works as CLIENT, multi connection, data transparent transmission.		
→	AT+CIPMUX=1	After CIPMUX=1, module can do transparent transmits only.
←	OK	
→	AT+CIPMODE=1	Transparent transmission mode
←	OK	
→	AT+CIPSTART=0,"TCP","172.168.1.119",4321	172.168.1.119 is an INTERNET server.
←	OK CONNECT OK	
→	AT+CIPSCON=1	Save parameter(connection parameter and transparent transmission mode)
←	OK	
	Reboot module. Reenter transparent transmission after reboot and set up two connections automatically. Connection parameter is the same as saved last time when +CIPSCON is input .	
1.4) Module works as SERVER, multi- connection, non-transparent transmission.		
→	AT+CIPMUX=1	
←	OK	
→	AT+CIPSERVER=1,8800	Module works as SERVER
←	OK	configuration is successful, Can monitor TCP and UDP link at the same time
→	AT+CIPSTATUS	
←	OK STATE:IP STATUS S:0,8800,LISTENING C:0,"","",,"",,"INITIAL" C:1,"","",,"",,"INITIAL" C:2,"","",,"",,"INITIAL" C:3,"","",,"",,"INITIAL"	
←	0,CONNECT OK	CLIENT link to module (TCP)，linkid = 0
←	1,CONNECT OK	CLIENT link to module (UDP)，linkid = 1 Note: No matter TCP or UDP, 4 links can be set up at most at the same time.
→	AT+CIPSTATUS	Inquire TCPIP level status
←	OK STATE:IP STATUS S:0,8800,LISTENING C:0,UDP,"192.168.4.1",8800,"192.168.4.112",65074,CONNECTED C:1,TCP,"192.168.4.1",8800,"192.168.4.112",50114,CONNECTED C:2,"","",,"",,"INITIAL" C:3,"","",,"",,"INITIAL"	
←	+RECEIVE,0,10： 0123456789	Receive data from CLIENT on link 0. the content is：0123456789
→	AT+CIPSEND=0,6	

←	>	Now data can be input. Input 6 characters, data will send initiatively.
←	0,SEND OK	Data sending successfully.
←	0,CLOSE OK	Opposite terminal（CLIENT） break connection initiatively.
→	AT+CIPSERVER=0	
←	OK we must restart	Need reboot module after shut down server.
1.5）Module works as SERVER, single and multi-connection, data transparent transmission.		
→	AT+CIPMUX=1	
←	OK	
→	AT+CIPSERVER=1,8800	Module works as SERVER
←	OK	Configuration is successful. can monitor TCP and UDP connection at the same time
→	AT+CIPMODE=1	
←	OK	
→	AT+CIPSCON=1	
←	OK	
	Will enter transparent transmission mode after reboot. And wait for CLIENT connection setup. There will be no reminder after connection set up successfully. Now data can be two-way transmitted.	module as server can monitor 4 TCP and/or UDP links at most.
2）Module operates in AP mode： The following words in light blue are the same common premise steps in this mode:		
→	AT+CWMODE?	Step 1: check WIFI level operation mode
←	+CWMODE:3 OK	Default mode is 3: AP+STA mode
→	AT+CWMODE=2	Step2: Set as AP mode
←	OK	
→	AT+RST	Step 3: Need reboot after setting mode.
←	OK	Note: Step2 and Step 3 can be ignored if after booting AT+CWMODE it is already Step 1.
→	AT+CWSAP="aptest","123456",1,0	Step4: Set mode to AP and configure AP parameters.
←	OK	**Note: After finishing this step, it will automatically save the original mode after module reboot. It will be AP also if it is AP mode before reboot, and keeps the same AP parameters (SSID, password and so on) as before reboot.**
	STATION join the aptest AP	
2.1）CLIENT，single connection, non-transparent transmission data：		
	Please refer to 1.1)	
2.2）CLIENT，single connection, transparent transmission data：		

	Please refer to 1.2)	
2.3）SERVER，multi-connection，non-transparent transmission data：		
	Please refer to 1.3) .the different part is all connections are LAN connection under this application.	
2.4) SERVER，multi- connection，non-transparent transmission data：		
	Please refer to 1.4)	
2.5) SERVER，single and multi-connection, transparent transmission data：		
	Please refer to 1.5)	
3）Module works in AP+STATION mode ： The parts in the following light blue grids are the common premise steps in this mode：		
→	AT+CWMODE=3	Set as AP + STA mode
←	OK	
→	AT+RST	Need reboot after configuring mode.
←	OK	
→	AT+CWJAP="yh_taihdhad","UUU02339"	Add into one AP which is a router can access outer net.
←	OK	
→	AT+CWSAP="aptest","123456",1,0	Configure module as AP（name as aptest），and configure AP parameter
←	OK	
	There is one computer(name as user) using WIFI to join this aptest AP.	
→	AT+CIPMUX=1	Start multi- connection mode
←	OK	
→	AT+CIPSTART=1,"TCP","120.209.197.145",8000	Set up TCP connection with one SERVER of INTERNET. Connection number is 1.
←	OK 1,CONNECT OK	Successfully connect to INTERNET SERVER.
3.1) Module works as CLIENT to link local server, meanwhile works as CLIENT to link INTERNET server.		
→	AT+CIFSR	
←	192.168.4.1 192.168.2.110 OK	192.168.2.110 is the allocated IP address when joins router which can surf outer net. 192.168.4.1 is the IP address and module works as AP itself.
→	AT+CIPSTART=2,"TCP","192.168.4.100",7777	To set up one connection with USER's server, connection number is 2.
←	OK 2,CONNECT OK	
←	+RECEIVE,2,11： hhhhhhfdddd	receive one pack of data from USER
→	AT+CIPSEND=1，11	Receive data from USER's server (link 2) and send to outer net's server (link 1)
←	>	input hhhhhhfdddd
←	1,SEND OK	When character's number is up to 11, it will send out by itself.

3.2) Besides linking to internet server, module also works as CLIENT to link local server, serve as multi- link and transparent transmission. :		
→	AT+CIPSTART=2,"TCP","192.168.4.100",7777	To set up one connection with USER's server, connection number is 2.
←	OK 2,CONNECT OK	
→	AT+CIPMODE=1	
←	OK	
→	AT+CIPSCON=1	
←	OK	
	Reboot module. Module will enter transparent transmission after reboot and automatically sets up 1 and 2 connections at the same time.	
→	+++	
→	AT+CIPSTATUS	Linking status can be checked after quiting transparent transmission mode
←	STATE:IP STATUS C:0,"","",,"",,"INITIAL" C:1,"TCP",”192.168.1.119”,12363,"120.209.197.145",8000,"CONNECTED" C:2,"TCP",”192.168.1.119”,36890,"192.168.1.100",7777,"CONNECTED" C:3,"","",,"",,"INITIAL"	
3.3）Besides linking to Internet server, module also works as server to monitor local connection, service including: multi- link and non- transparent transmission.		
→	AT+CIPSERVER=1,3366	
←	OK	
←	0,CONNECT OK	accept a link , linkid is 0.
←	2,CONNECT OK	accept a link , linkid is 2.
→	AT+CIPSEND=0,8	
←	>	After input 12345678, then data can send out by itself.
←	0,SEND OK	
→	AT+CIPSTATUS	Inquire TCPIP status
←	OK STATE:IP STATUS S:0,3366,LISTENING C:0,UDP,”192.168.4.1”,3366,”192.168.4.100”,51733,CONNECTED C:1,TCP,”192.168.1.119”,35669,”120.209.197.156”,8000,CONNECTED C:2,TCP,”192.168.4.1”,3366,”192.168.4.100”,57957,CONNECTED C:3,"","",,"",,"INITIAL"	
3.4）Besides linking to Internet server, module also works as server to monitor local connection ,		

service including: multi- link and transparent transmission.		
→	AT+CIPSERVER=1,3366	
←	OK	
→	AT+CIPMODE=1	
←	OK	
→	AT+CIPSCON=1	
←	OK	
	Reboot module. It will enter transparent transmission mode automatically and set up connection 1,"TCP","120.209.197.145",8000. Module works as SERVER at the same time to monitor port 3366, waiting for setting up connection from client	